Trajectory Tracking Control of Pendubot Using the Takagi-Sugeno Fuzzy Scheme.

Zhen Cai

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

March 2002

0-612-68450-4

Abstract

Trajectory Tracking Control of Pendubot Using the Takagi-Sugeno Fuzzy Scheme

Zhen Cai

Pendubot is a planar two-link underactuated robotic mechanism and serves as a benchmark example to test developed control algorithms for underactuated mechanical systems. This device is a two-link planar robot with an actuator at the link 1 and no actuator at the link 2. The control difficulty for the Pendubot is obviously due to the reduced dimension of the input space. The literature on the control of the Pendubot is mainly to stabilize this system in its vertical position. However, if the trajectory tracking for the Pendubot is required, the controller design is a challenging task since the Pendubot is not right invertible. The literature on the trajectory tracking is spare.

In this thesis we attempt to use the Takagi-Sugeno fuzzy model for the task of the trajectory tracking of the Pendubot. With this fuzzy approach, each fuzzy local model represents a linearized model of the operation point of the Pendubot. Two proposed control algorithms employ the fuzzy control that is designed by an aggregation of the fuzzy local controllers. For the first control algorithm, the local controller consists of an optimal output feedback plus a term for perturbation rejection, which is design based on the output optimal control and the linear regulatory theory. For the second control algorithm, the local controller is constructed with the state observer. The advantage for such designs is only simple fuzzy controllers are used in the approaches. The proposed two control laws ensure global stability of the closed-loop system and the first one also guarantees the optimal output trajectory tracking.

To verify the proposed schemes, real time experiments are conducted for the first control algorithm in Pendubot. By forcing the Pendubot to track a sinusoid signal of significative amplitude in real time around an unstable equilibrium point, the experimental results illustrate the applicability of the proposed algorithm.

Acknowledgements

The author dedicates this thesis to his thesis supervisor Dr. Chun-Yi Su, whose invaluable guidance and knowledge has made this thesis a success and I look forward to his guidance in all future endeavors.

Special thanks go to Mr. Feng Zhao who built the system and helped perform the experiments during the development of this thesis. His ideas and skilled hands were invaluable while building the Pendubot and are greatly appreciated.

Secondly, I would like to thank Mr. Tianshun Lu who gave me a lot of help during my thesis work.

Finally, I would like to thank my entire family for their love and support throughout my graduate school days. They gave me the encouragement to complete this work.

Table of Contents

List of Figures

Nomenclature

$D(q)$: Inertia Matrix

$C(q,\dot{q})$: Coriolis and Centrifugal Matrix

$G(q)$: Gravitational Vector

$F(q)$: Friction Vector

τ : input torque

q_1 angle of link 1

$\dot{q}_1$: velocity of link 1

q_2 : angle of link 2

$\dot{q}_2$: velocity of link 2

$\lambda(z(t))$: membership function

P_1 : Positive Definite Matrix

X : a neighborhood of the origin of R^n

Z : a neighborhood of the origin of R`

W : a neighborhood of the origin of R`

V : Lyapunov Function

σ : the spectrum

Chapter 1

Introduction

1.1.Overview

Control of under-actuated mechanical systems or robots represents an important class of control issues that are becoming increasingly important [1]-[3]. Examples of such systems are illustrated in [4]-[7]. Frequently cited applications include saving energy and reducing weight; these applications are attained by using fewer actuators and by gaining fault tolerance to actuator failures. Interest in studying the under-actuated mechanical systems is also motivated by their role as a class of strong, nonlinear systems where complex internal dynamics, nonholonomic behavior, and lack of feedback linearizability are often exhibited. Traditional nonlinear control methods cannot address the challenges and new approaches must be developed; this will stimulate new results in robot control theory in the near future.

Though the dynamics of the under-actuated mechanical system are well understood, the difficulty of solving the control problem for underactuated mechanisms is obviously due to reduced input space. The literature on the control of underactuated systems has recently been written in [6] [8] [9], and discussion focuses on two degree of freedom examples [10] [11]. Earlier works done in this field deal with the control of under-actuated robotic systems in [12] and [13]. Underactuated mechanical systems have also been investigated from nonholonimic constraint point of view [14] [15]. Take for instance the study of Oriolo, Nakamura and Wichlund *et al* [16] who established the conditions for partial integrablity of second-order nonholonomic constants and discussed control problems.

While some interesting techniques and results have been presented in the abovementioned publications, the control of such systems still remains a problem. For example, most of the control schemes mentioned above either failed to provide a thorough analysis of the overall system stability, or assumed that gravitational forces do not act on the passive joints. Furthermore, a precise knowledge of the dynamic model is generally required.

The Takagi-Sugeno (T-S) fuzzy methodology [17] has been used to model nonlinear systems. One of the principal reasons to employ this technique is the fact that the dynamics of a nonlinear system are easily obtained by the linearization of nonlinear models near different operation points or by the input and output identification around these points. Once the linear models are obtained, the linear control methodology can be used to design controllers for each local model. The overall controller would be a fuzzy blending of the local controllers. Several algorithms have been developed in recent years due to the research that has been done on the problem of the set-point regulation. The work in [18] shows that stability of the T-S fuzzy control system can be established by finding a common positive definite matrix that satisfies the associated Lyapounov equations; furthermore, an algorithm to obtain a stable closed loop fuzzy T-S system is given. Basically it proposes, as a local linear controller, a state feedback law with the gains selected to stabilize each local system and then the stability of the overall fuzzy system is tested. If stability is not achieved, the local gains must be recalculated. This iterative process concludes when the overall system is stable but to avoid this iterative process, an optimal fuzzy controller design via convex optimization techniques based on LMIs is presented in [19]. There, the upper bound of a given quadratic performance function is minimized for the deterministic continuous-time case.

Nonlinear system trajectory tracking is more difficult than set-point regulation and has seldom been analyzed. In [20], tracking is achieved by minimizing the error between the nonlinear system and a nonlinear reference model; both of them are modeled using T-S methodology. The error is then linearized and closed loop stability conditions are established by means of LMIs. In the work [21], a fuzzy control scheme combining linear regulatory theory [22] [23] with the T-S fuzzy control methodology is proposed. However, this scheme lacks stability analysis.

In this thesis, we will use the T-S fuzzy model to model a class of nonlinear systems. Each fuzzy local model represents a linearized model of the operation point of the controlled nonlinear system. The proposed control algorithm employs the fuzzy control designed by an aggregation of the fuzzy local controllers. The local controller for each local model consists of an optimal state feedback plus a term for perturbation rejection based on linear optimal control and linear regulatory theory. The proposed control law ensures global stability of the closed loop system and guarantees the optimal tracking control. As a more realistic and common approach than the state feedback controller, the fuzzy output controller that only utilizes the tracking error is also proposed. For each local model, it is not necessary to assume that the state is completely measured and only the error e is available for measurement. The stability of the entire closed-loop fuzzy system is ensured by lyapunov stability analysis.

The PENDULUM ROBOT, as a benchmark underactuated system, (Pendubot [2]) is an electro-mechanical (or Mechatronic) system consisting of two rigid links interconnected by revolute joints. The first joint is actuated by a DC-motor and the second joint is underactuated. Thus, the second link may be thought of as a simple pendulum whose motion can be controlled by actuation of the first link. The Pendubot is similar in spirit to the classical inverted pendulum on a cart. Using the Pendubot, one can investigate not only the problem of the set-point regulation, including swinging up and balancing, but also trajectory tracking.

To verify these proposed schemes, experiments are conducted with the Pendubot. At first, we will test the controllers for swing up and balance. For trajectory tracking, we will force the Pendubot to track a sinusoid signal of significative amplitude in real time around an unstable equilibrium point.

1.2. Objectives

The objective of this thesis is to develop fuzzy controllers for trajectory tracking of nonlinear systems. Important issues for consideration are development of fuzzy logic algorithms and their implementation. Application of fuzzy controlling methods to actual robots is the motivation of this study. Implementation will occur in real time to verify the advantages of fuzzy control.

The Pendubot is installed in the Robotics and Mechatronics Laboratory, led by Dr.Su. It is a typical under-actuated pendulum robot which is an electro-mechanical system consisting of two rigid links interconnected by revolute joints.

1.3. Thesis Layout

In chapter two we explain the system of the Pendubot. The components of the Pendubot are first described and we will quickly go through the derivation of the mathematical model of the Pendubot. The ordinary differential equations (O.D.E.s) found in this section are the basis for the controller designs used.

The next chapter discusses some classical control algorithms used to swing up and balance the links at unstable equilibrium points. For the swing up control the partial feedback linearation method is used as discussed in [2] and [3]. The balancing control was then found by linearizing the system and designing a full state feedback controller for that linearized model. The combined algorithm was then used to link these two control algorithms by observing the states.

Chapter four proposes a very interesting algorithm for optimal trajectory tracking. In this chapter, the T-S fuzzy methodology is used to model a class of nonlinear systems. Each fuzzy local model represents a linearized model of the operation point of the controlled nonlinear system. The proposed control algorithm employs the fuzzy control that is designed by an aggregation of the fuzzy local controllers. The local controllers for each local model consist of an optimal state feedback plus a term for pertubation rejection that was designed based on output optimal control and linear regulatory theory. The advantage for such a design is only a simple fuzzy controller is used in the approach. The stability of the entire closed-loop fuzzy system is ensured by the lyapunov stability analysis.

In chapter five, experiments are conducted to verify the proposed algorithm. At first, we will swing the pendubot to the unstable equilibrium point and then force the pendubot to track a desired trajectory. The results of the implemented controllers are displayed.

In chapter six, a similar control algorithm using output feedback for achieving trajectory tracking is proposed. The proposed control law ensures global stability of the closed-loop system and guarantees the asymptotic output tracking.

Chapter seven displays the simulation result of the proposed algorithm of chapter six.

Summary, conclusions, the recommendations for future work, reference and publications are presented afterwards. Finally, Appendix A gives the linearization equations. Appendix B lists the Maple Files to calculate the PI and GAMA. Appendix C gives the Matlab M-Files used to calculate the gain K and L. Appendix D is the parameters for the local linear system and linear gain. Appendix E lists the source code for trajectory tracking around the top equilibrium point. Appendix F lists the source code for trajectory tracking around the mid equilibrium point. Appendix G is the parameters for the local linear system and observers. Appendix H is the Matlab Files used to calculate the parameters of the local linear observers. Appendix I lists the source code for the simulation.

1.4. Contributions

The approach in this thesis is in someways partly similar to the work of [2] [21], but there are significant differences as well, which are the personal contributions of the author.

1) A simple fuzzy controller combining linear optimal theory and linear regulator theory with the T-S method is proposed for the optimal trajectory tracking of nonlinear systems. Another fuzzy control algorithm in which only the position signal is available is also proposed as an alternate approach for output tracking of nonlinear systems. This control algorithm is more common than the algorithm assuming all states are observable.
2) Lyapunov function is used to verify the stability of the entire closed-loop system. In some papers [18] [33], a common positive definite matrix must be selected to make every local system satisfy the linear matrix inequalities (LMI), but by the stability

analysis in the section 4.3 and section 6.3. the common positive matrix is not necessary. If we can make every local system stable. the overall system is stable.

3) The first algorithm is implemented in real time. In [2], it was thought that a 40^0 offset from the top and mid positions was the limit for the motor balancing the links, but in this thesis. the Pendubot is forced to track a significative signal of 70^0. It can be thought the main advantage of the fuzzy control.

4) A simulation example is given to illustrate the designed procedures and tracking performances of the second control algorithm. The control effort is excellent and without control saturation.

Chapter 2

Pendubot System Description

2.1. Mechanical Design and Controller Interface

The Pendubot scheme is illustrated in Figure 2.1. The actuated joint is driven by a high torque 90VDC permanent magnet DC-motor without gear reduction. To give joint one direct

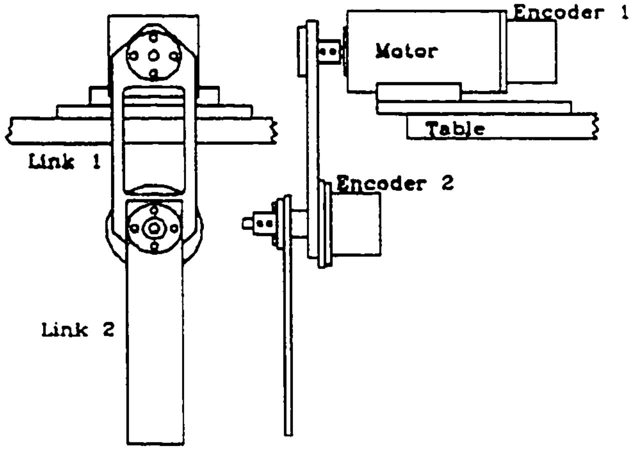

Fig 2.1 Front and Side Perspective Drawings of the Pendubot

control, we designed the Pendubot to hang off the side of a table with a coupling link attached directly to the shaft of the motor. The mount and bearings of the motor are the support for the entire system. Link one also includes the bearing housing which allows for the motion of joint two. Needle roller bearings riding on a ground shaft were used to construct this revolute joint two. The shaft extends out of both directions of the housing, allowing the coupling to attach both link two and an optical encoder mounted on link one. This optical encoder produces the position feedback of link two. The design gives both links 360^{0} of motion. Link one, however, cannot continously rotate due to the encoder cable for link two. Link two has no constraints on continuous revolutions.

Link two is a ¼ inch (0.635 cm) length of aluminum with a coupling that attaches to the shaft of joint two. It is 9 inches (22.86- cm) long. Link 1 is 6 inches (15.24-cm) long and the length of links was determined by [2].

The final component of the Pendubot's hardware is its controller. See Figure 2.2 for a pictorial description of the interface between the Pendubot and the controller. BEI 1024 counts/rev resolution optical encoders, one attached at the elbow joint and one attached to the motor, were used as the feedback mechanism for the joint angles. Advanced Motion Control's 25A8 PWM servo amplifer was used to drive the motor. In the control algorithm the amplifier can be thought of as just a gain. In the case of the Pendubot, the amplifier was set in torque mode and adjusted if for a gain of 1V=1.2 Amps.

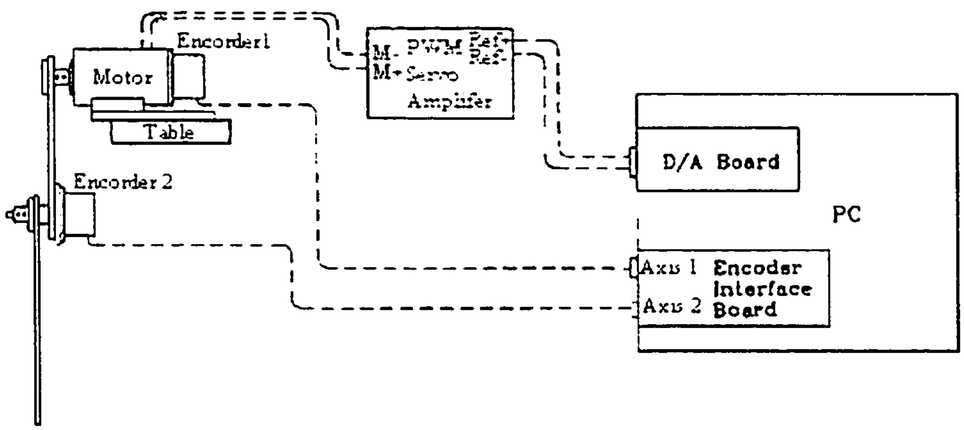

Figure 2.2 Pictorial of the Pendubot's interface with its controller

In an attempt to simplify the controller for the Pendubot and minimize its cost, the control algorithm was implemented using only the microprocessor in our PC instead of purchasing an additional DSP card. A Pentium125 IBM compatible PC with a D/A card and an encoder interface card was used. The DAC-02 card by Keithley Metrabyte was used for the digital to analog converter and the 5312 B by Technology 80 was used to interface with the optical encoders. The X4 quadrature mode was used on this card to increase the resolution of the optical encoders by 4, thereby giving 4096 counts/rev. Then with the software library routines accompanying the cards, the control algorithms were implemented.

The only difficulty in using a PC as a digital controller is finding a way to get a reliable fast sampling period interrupt. DOS does produce a clock pulse; but it only occurs every 55 milliseconds making it useless for this system which needs at least a 10 ms sampling period. To bypass this, the time chip on the motherboard of the PC was programmed directly to achieve a higher resolution. The software package "PC Timer Tools" by Ryle Design includes an alarm algorithm that can be used to produce an appropriate sampling period (i.e. 5ms). The format of the control algorithm is as follows:

```
/*Perform all needed initializations*/
/*start 5ms alarm*/
while(Continue_Control= =TRUE){
/*sample encoder positions*/
/*use definite differences to calculate velocity*/
/*calculate needed control effort*/
/*output control value to motor*/
while (Alarm_expired= = FALSE){
/*continue to loop until alarm expires*/
}/***** end of second while*****/
}/*****end of first while ****/
```

This control design worked very well. A reliable 1ms sampling period was achieved even when computing the inverse dynamic equations for the partial feedback linearization control.

A 5 ms sampling period was used by most of our controllers in order to allow the saving of response data while the controller was operating.

2.2. Control System Design

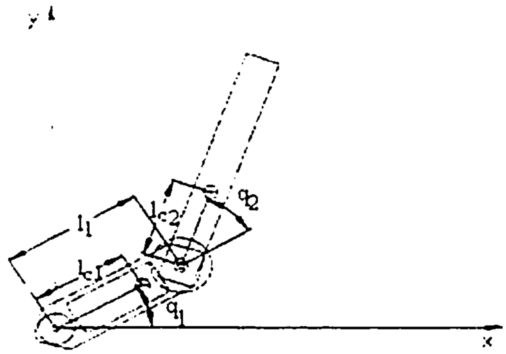

Fig.2.3. Coordianate Description of the Pendubot. l_1 is the length of link one. l_{c1} and l_{c2} are the distances to the center of mass of the respective links. and $q_1$and q_2 are the joint angles of the respective links

The dynamic matrix equation of motion for the Pendubot is given by [24]

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \qquad (2.1)$$

where $\tau = [\tau_1 \quad 0]^T$ is the vector of torque applied to the links and q is the vector of joint angle positions with

$$D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \qquad (2.2)$$

$$d_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2) + I_1 + I_2$$

$$d_{12} = d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2$$

$$d_{22} = m_2 l_{c2}^2 + I_2$$

and

$$C(q,\dot{q}) = \begin{bmatrix} h\dot{q}_2 & h\dot{q}_2 + h\dot{q}_1 \\ -h\dot{q}_1 & 0 \end{bmatrix} \tag{2.3}$$

$$h = -m_2 l_1 l_{c2} \sin q_2$$

$$G(q) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \tag{2.4}$$

$$\phi_1 = (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2)$$

$$\phi_2 = m_2 g l_{c2} \cos(q_1 + q_2)$$

$$F(\dot{q}) = \begin{bmatrix} F_1(\dot{q}) \\ F_2(\dot{q}) \end{bmatrix} = \begin{bmatrix} k_{\mu 1}\dot{q}_1 \\ k_{\mu 2}\dot{q}_2 \end{bmatrix}$$

where

l_1 is the length of link 1

l_{c1} is the distance of the center of mass of link 1.

l_{c2} is the distance of the center of mass of link 2.

m_1 is the total mass of link 1.

m_2 is the total mass of link 2.

I_2 is the moment of inertia of link one about its centroid.

I_1 is the moment of inertia of link two about its centroid.

$k_{\mu 1}$ is the friction constant of link 1.

$k_{\mu 2}$ is the friction constant of link 2.

From the above equations it is observed that except friction constants, the seven dynamic parameters can be grouped into the following five parameters.

$$
\begin{aligned}
\theta_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + I_1 \\
\theta_2 &= m_2 l_{c2}^2 + I_2 \\
\theta_3 &= m_2 l_1 l_{c2} \\
\theta_4 &= m_1 l_{c1} + m_2 l_1 \\
\theta_5 &= m_2 l_{c2}
\end{aligned}
\tag{2.5}
$$

Using the Energy Equation Method of System Identification [2], the above parameters are

$$
\begin{aligned}
\theta_1 &= 0.014806\,Kgm^2 \\
\theta_2 &= 0.005096\,Kgm^2 \\
\theta_3 &= 0.00456684\,Kgm^2 \\
\theta_4 &= 0.0010033\,Kgm \\
\theta_5 &= 0.030285\,Kgm
\end{aligned}
$$

The friction constants is:

$$
k_{d1} = 0.00545\,Kgm^2/s \qquad k_{d2} = 0.00047\,Kgm^2/s
$$

These five parameters are all that are necessary for a control design that neglects friction. There is no reason to go a step further and find the individual parameters since the control equations can be written with only the five parameters. Substituting these parameters into above equations leaves the following matrices.

$$
D(q) = \begin{bmatrix} \theta_1 + \theta_2 + 2\theta_3 \cos q_2 & \theta_2 + \theta_3 \cos q_2 \\ \theta_2 + \theta_3 \cos q_2 & \theta_2 \end{bmatrix} \tag{2.6}
$$

$$
C(q,\dot{q}) = \begin{bmatrix} -\theta_3 \sin(q_2)\dot{q}_2 & -\theta_3 \sin(q_2)\dot{q}_2 - \theta_3 \sin(q_2)\dot{q}_1 \\ \theta_3 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \tag{2.7}
$$

$$
G(q) = \begin{bmatrix} \theta_4 g \cos q_1 + \theta_5 g \cos(q_1 + q_2) \\ \theta_5 g \cos(q_1 + q_2) \end{bmatrix} \tag{2.8}
$$

Finally, using the invertible property D(q) [24], we obtain:

$$
\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = D(q)^{-1}\tau - D(q)^{-1}C(q,\dot{q})\dot{q} - D(q)^{-1}G(q) - D(q)^{-1}F(\dot{q})
$$

The states are selected as:

$$x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2, x_4 = \dot{q}_2$$

The state equations are:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \ddot{q}_1 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \ddot{q}_2
\end{aligned}$$

$$\dot{x}(t) = f(x) + g(x)u(t) = F(x(t), u(t), t) \tag{2.9}$$

Chapter 3

The Pendubot Classical Control Algorithm

3.1 Swing up control

As stated earlier, the goal for the Pendubot controller is to swing the links form their hanging position to an unstable equilibrium position and then balance the links at equilibrium. This control is divided into two parts, swing up control, and balancing control or trajectory tracking control. The swing up control uses the partial feedback linearization method. Many different control algorithms could have been used to perform the swing up. In fact, a PID controller positioned at link one for the upswing Pendubot worked well but amplified numerical noise. Alternatively, Partial Feedback Linearization needs position feedback for both link one and link two, but takes into account the nonlinear facts of the linkage. For an illustration of the general derivation of partial feedback linearization, please refer to [25] and [26].

The equations of motion of the Pendubot are given by equation (3.1). Neglecting the friction constants and performing the matrix and vector multiplications, the equations of motion are written as:

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + \phi_1 = \tau_1 \tag{3.1}$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + \phi_2 = 0 \tag{3.2}$$

Due to the underactuation of link two, we cannot linearize the dynamics of both degrees of freedom. However, one of the degrees of freedom can be linearized. This allows for the design of an outer loop control that tracks a given trajectory for the linearized degree of freedom. In the case of the Pendubot, linearization about the collocated degree of freedom q_1 was chosen Equation (3.2) was solved for the angular acceleration of link two.

$$\ddot{q}_2 = \frac{-d_{21}\ddot{q}_1 - c_{21}\dot{q}_1 - \phi_2}{d_{22}} \tag{3.3}$$

This was then substituted into equation (3.1) and written as

$$\bar{d}_{11}\ddot{q}_1 + \bar{c}_{11}\dot{q}_1 + \bar{c}_{12}\dot{q}_2 + \bar{\phi}_1 = \tau_1 \tag{3.4}$$

with

$$\begin{aligned} \bar{d}_{11} &= d_{11} - \frac{d_{12}d_{21}}{d_{22}} \\ \bar{c}_{11} &= c_{11} - \frac{d_{12}c_{21}}{d_{22}} \\ \bar{c}_{12} &= c_{12} \\ \bar{\phi} &= \phi_1 - \frac{d_{12}\phi_2}{d_{22}} \end{aligned} \tag{3.5}$$

Then, just as with the full linearization method (also called the computed torque method [24]) the inner loop control that linearizes q_1 can be defined as

$$\tau_1 = \bar{d}_{11}v_1 + \bar{c}_{11}\dot{q}_1 + \bar{c}_{12}\dot{q}_2 + \bar{\phi}_1 \tag{3.6}$$

This results in the system

$$\ddot{q}_1 = v_1 \tag{3.7}$$

$$d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + \phi_2 = -d_{21}v_1 \tag{3.8}$$

Since the equation is now linear, an outer loop control can be designed to track a given trajectory for link one. The response of link two then is given by the resulting nonlinear equation. This equation represents internal dynamics with respect to an output $y=q_1$. The goal of the outer loop control is to track a given trajectory for link one and at the same time excite the internal dynamics to swing link two to a balancing position. A PD with feedward acceleration was chosen for application in the Pendubot.

$$v_1 = \ddot{q}_1^d + K_d\left(\dot{q}_1^d - \dot{q}_1\right) + K_p\left(q_1^d - q_1\right) \tag{3.9}$$

See Figure 3.1 for a block diagram of the swing up control.

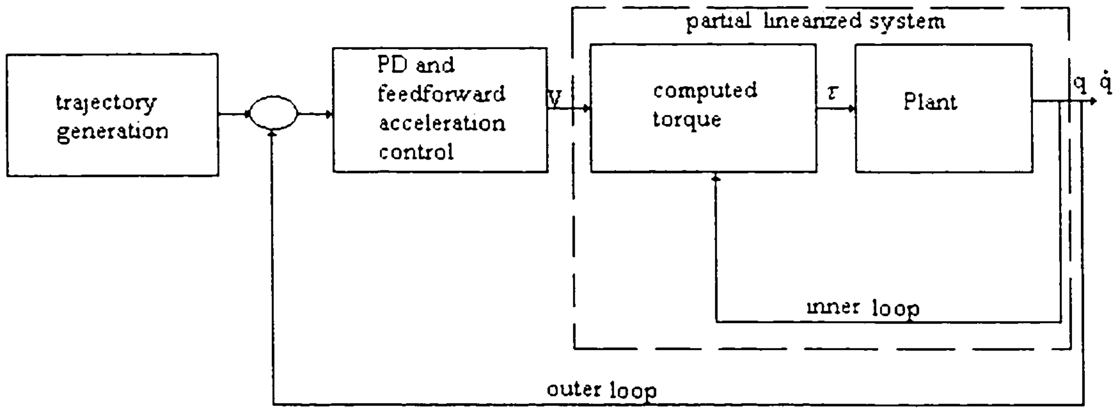

Fig 3.1 Block Diagram of the Partial Feedback Linearization Control

With this controller, a trajectory must be determined to swing the links to their unstable equilibrium position. To swing the links to the upright top position ($q_1=\pi/2$, $q_2=0$), we used the step trajectory $q_1=\pi/2$. The swing up trajectory to swing the links to the mid position ($q_1=-\pi/2$, $q_2=\pi$) was somewhat more difficult. The trajectory for the mid position can be written as:

$$q_1 = 1.4\sin(5t) - \pi/2 \qquad t < 2\pi/5$$
$$q_1 = -\pi/2 \qquad t > 2\pi/5$$

This trajectory pumps energy into the system by causing the second link to swing back and forth and finally up to its middle equilibrium point.

To fine tune the swing up control, the K_p and the K_d gains were adjusted. Corrected gains were found that swing the second link slowly into its equilibrium positions so that another controller can catch the link.

3.2 balancing control

The control for balancing the Pendubot is very similar to the classical cart-pole inverted pendulum problem. To design the controller, we linearized the Pendubot's nonlinear equations of motion (2.9), and designed a full state feedback controller with the linear model. The Taylor series approximation

$$f_d(x,u) = f_d(x_r,u_r) + \left.\frac{\partial f}{\partial x}\right|_{x_r,u_r}(x - x_r) + \left.\frac{\partial f}{\partial x}\right|_{x_r,u_r}(u - u_r) \tag{3.10}$$

was used to linearize the plant. In equation (2.9), x is the vector of states, u is the single control input for the Pendubot and x_r and u_r are the equilibrium values of the states and control. Since the interest is only in controlling the Pendubot at equilibrium positions, $f_d(x_r,u_r)$ will always be zero. All that is needed then is to find out the partial derivative matrices and evaluate them at the equilibrium points. Studying the equations (2.6) though (2.9) shows that the Pendubot's equilibrium points can be defined by:

$$u_r = \theta_4 g \cos(x_{r1}) \tag{3.11}$$

$$x_{r1} + x_{r3} = \pi / 2. \tag{3.12}$$

Differentiating equation with respect to the states leaves the A matrix in the linearized model:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \tag{3.13}$$

The B matrix is found by the partial derivative with respect to the control input:

$$\frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix} \tag{3.14}$$

Refer to Appendix A for a full derivation of these partial derivative terms.

Each equilibrium point defines a different linearized system (See Appendix A). This signifies that different control gains will be needed for each equilibrium point for best results in balancing the Pendubot. We define the top balancing position as the upright position with $x_{r1}=\pi/2, x_{r3}=0$ and $u_r=0$. The middle balancing position is defined as $x_{r1}=-\pi/2$, $x_{r3}=\pi$ and $u_r=0$.

Using these equilibrium values and the parameters identified by the energy equation method, the linear models for the top and mid equilibrium positions are as follows:

Top

$$\dot{x} = Ax + Bu$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 66.9725 & -0.2445 & -24.8339 & 0.04 \\ 0 & 0 & 0 & 1 \\ -68.7259 & 0.4637 & 105.3692 & -0.1202 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 44.8715 \\ 0 \\ -85.0866 \end{bmatrix}$$

Mid

$$\dot{x} = Ax + Bu$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -66.9725 & -0.2445 & 24.8389 & 0.0022 \\ 0 & 0 & 0 & 1 \\ 65.219 & 0.02538 & 55.6914 & -0.0446 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 44.8715 \\ 0 \\ -4.6565 \end{bmatrix}$$

With these linear models the LQR or pole placement techniques can be used to design full state feedback controllers. For example, let assume q_2 is output and $y=Cx$, and the cost function is:

$$J = \int_0^\infty \left(y^T \Lambda y + u^T Ru\right)dt$$

with

C=[0 0 1 0]

R=[1].

$$\Lambda = \begin{bmatrix} 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The state feedback control law u=-Kx is designed to minimizes the cost function.

Fig. 3.2 and Fig. 3.3 are the photos balancing the link at the equilibrium position.

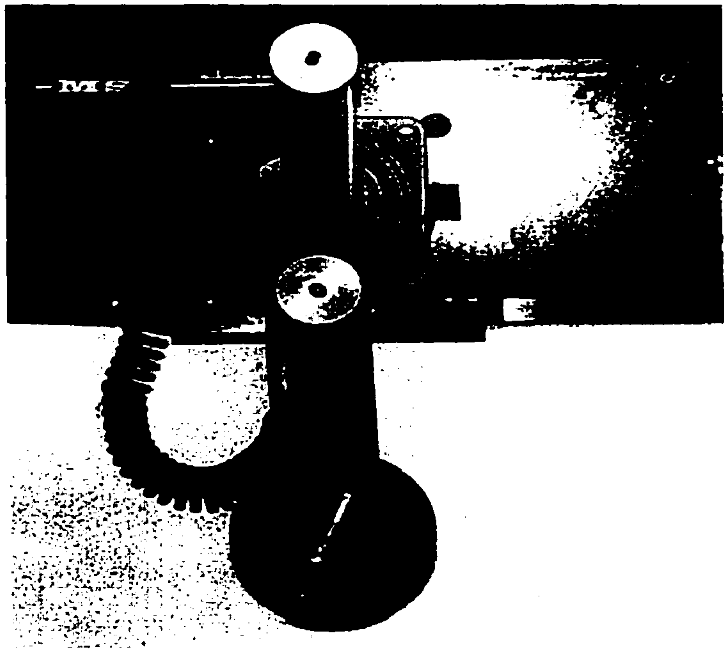

Fig 3.2 Pendubot in the Mid Position

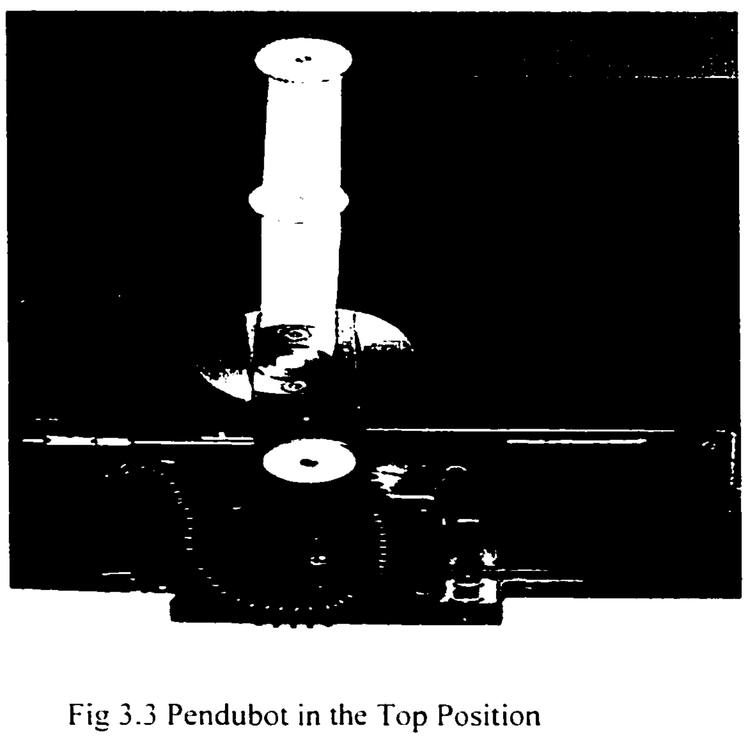

Fig 3.3 Pendubot in the Top Position

The top and middle positions are not the only possible equilibrium positions of the Pendubot. In fact there is a continuum of positions. Figure 3.4a shows another possible configurations. There are four uncontrollable positions, $x_{r1}=0, x_{r3}=\pi/2$ or $-\pi/2$ and $x_{r1}=-\pi$, $x_{r3}=\pi/2$ or $-\pi/2$. Figure 3.4 b shows the first of these uncontrollable positions.

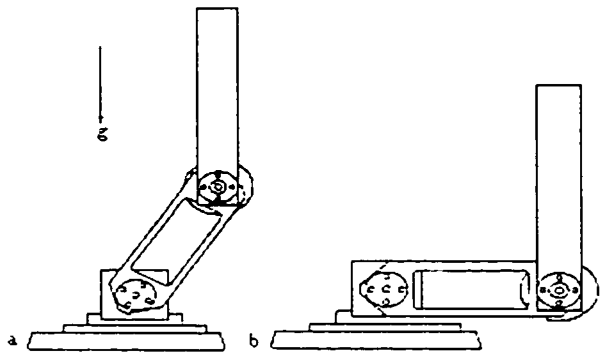

Fig 3.4 Other possible Equilibrium Positions. a. Another controllable Positions. b. One of the four uncontrollable equilibrium.

In the following chapters, a new control algorithm will be used to force the Pendubot to track a desired trajectory. To realize it in real time, other equilibrium positions will be used. In [2], the author believes that the Pendubot would become unstable in the approximately 40^0 offset from the top and mid positions. But, because of fuzzy algorithm's excellent control, the Pendubot can be made stable in the approximate 70^0 offset. This is the main contribution of the thesis.

3.3 Combining and implementing the controllers

3.3.1 The Combining Algorithm

With both the swing up control and balancing control complete, an algorithm was needed to connect the two. The following algorithm was used to give more intelligence to the Pendubot and switch the control by observing the states of the system.

```
if |x1 - xr1| < .10   {
 if |x3 - xr3| < 0.20{
   u=-Kx;
   if |u| < 9/*Volts*/{
    /*Output balancing control*/
  }else/*Output swing up control*/
 }else/*Output swing up control*/
}else/*Output swing up control*/
```

This algorithm waits for link one to arrive within 0.1 radians of its equilibrium position and then checks link two. If link two is also within 0.20 radians of its equilibrium position, the balancing control is calculated. If the control output is less than 9 volts, 10 volts being the maximum DAC output for the Pendubot, the control is switched to the balancing mode. Otherwise the links are passing too quickly though the equilibrium point and the swing up control remains intact.

3.3.2. joint velocities

Another implementation issue that arises in the controller design of the Pendubot is the approximation of the joint velocities. There is only one position feedback in the system so a finite approximation is used to estimate the velocity. To find the velocity we simply used the finite difference method. [x(k)-x(k-1)]/sample period was used. This creates numerical error or noise in the calculation of the control effort, though, due to the finite resolution of the optical encoders. Taking the average of the last three velocities helped to filter and decrease this noise. For example, the velocity calculation for joint one, state $x_2(k)$, is written as follows

$$x_2(k) = \frac{x_1(k) - x_1(k-1)}{\Delta t}$$

$$x_2(k) = \frac{x_2(k) + x_2(k-1) + x_2(k-2)}{3}$$

A dither signal was also needed to help balance the links in the top position. Due to friction and the increased effect of gravity on the links in the top position, the balancing control was not able to hold the links motionless. The balance control at the mid position did not have this problem. The main reason for this is that gravity works with the control at the mid position to keep link one at equilibrium. Whereas with the top position, gravity works to pull both of the links away from the equilibrium. The motion produced was a large amplitude away (approximately 0.2 radians). To eliminate some of this motion an open-loop dither signal was added to the control.

$$U=-Kx+0.25\sin 40.0t.$$

This dither signal helped to eliminate much of the sway but it was not able to cancel all of the motion.

Chapter 4

Optimal Fuzzy Trajectory Tracking Algorithm

4.1. T-S Fuzzy Methodology

In this methodology, nonlinear systems are approximated by a set of linear local models. A dynamic Takagi-Sugeno fuzzy model [17] is described by a set of fuzzy "IF-THEN" rules, with fuzzy sets in the antecedents and local linear time invariant systems in the consequents. Every rule of a T-S fuzzy model has the following form:

Plant Rule i

IF $z_1(t)$ is M_{1i}... $z_j(t)$ is M_{ji} .... $z_q(t)$ is M_{qi}

$$\text{Then}\begin{cases}\dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t)\end{cases}$$

where i=1..... r with r is the number of rules; the z_j (j=1....q) are the premise variables, which may be functions of the states of other variables; the M_{ji} are the fuzzy sets; $x \in R^n$ is the state vector, $u \in R^m$ is the input vector; $y \in R^r$ is the state output vector; A_i, B_i, C_i, are the matrices of adequate dimensions and $z(t)$ is the vector containing all the z_j.

The final state and output of the fuzzy system is inferred as follows:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))(A_i x(t) + B_i u(t))}{\sum_{i=1}^{r} \lambda_i(z(t))}$$

$$y(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))C_i x(t)}{\sum_{i=1}^{r} \lambda_i(t)}$$

4.2. Linear Regulator and Optimal Control theory

Consider a linear system

$$\dot{x}(t) = Ax(t) + Bu(t) + P\omega(t)$$
$$\dot{\omega}(t) = S\omega(t)$$
$$e(t) = Cx(t) + Q\omega(t)$$

where x is the internal state vector of the plant and defined on a neighborhood X of the origin of R^n; u is the control input vector: ω is a vector containing external disturbances and or references and defined on a neighborhood W of the origin of R^s. And e is the tracking error.

In linear optimal control theory, if $\omega(t) = 0$ then the control goal is to obtain the optimal gain matrix K, so that the feedback law U=Kx minimizes the cost function:

$$J = \int_0^{\infty} \left(e^T \Lambda e + u^T R u\right) dt$$

where Λ, R is positive definite.

Let F be the unique positive definite solution to the associated matrix Riccati equation:

$$0 = FA + A^T F - FBR^{-1}B^T F + C^T \Lambda C$$

where the optimal control gain can be defined as K= $B^T F$

When disturbances and references $\omega(t) \neq 0$, linear regulator theory [22], [23], can be utilized to ensure the asymptotic output tracking. In linear regulator theory, the control goal is to obtain a stable closed loop system and asymptotic tracking error for every possible exogenous input in a prescribed family of functions of time. This latter requirement is also known as the property of *output regulation*.

To ensure $e(t) \rightarrow 0$, the following is assumed:

A1. The exosystem is antistable: all the eigenvalues of S have nonnegative real parts.

Assuming (A1), the problem of output regulation via state feedback can be determined if, and only if, there exists matrices Π and Γ which solve the following linear matrix equation:

$$\begin{aligned} \Pi S &= A\Pi + B\Gamma + P \\ 0 &= C\Pi + Q \end{aligned} \tag{4.1}$$

The control law is given as,

$$\begin{aligned} u(t) &= Kx(t) + (\Gamma - K\Pi)\omega(t) \\ &= Kx(t) + L\omega(t) \end{aligned}$$

where $K=B^{1}F$, $L = \Gamma - K\Pi$ (4.2)

4.3. Proposed Algorithm

At this stage, a new T-S fuzzy control for nonlinear system trajectory tracking is presented. To design control rules, the same fuzzy sets used in antecedents are used in plant rules. Consequents, local laws based on the linear regulator theory and linear optimal control theory are designed for each local linear model. The rules for the plant and controller are:

i^{th} **Plant rule**

IF $z_1(t)$ is M_{i1}, ... , $z_j(t)$ is M_{ij}, ..., $z_q(t)$ is M_{qi},

$$\text{Then } \begin{cases} \dot{x} = A_i x + B_i u + P_i\omega \\ \dot{\omega} = S\omega \\ e = C_i x + Q_i\omega \end{cases} \tag{4.3}$$

i^{th} **Controller rule**

IF $z_1(t)$ is M_{ij}, ... , $z_j(t)$ is M_{ij}, ..., $z_q(t)$ is M_{qi}

$$\text{Then } u = K_i x(t) + L_i \omega(t) \tag{4.4}$$

where $K_i = B_i^T F_i$, $L_i = \Gamma_i - B_i^T F_i \Pi_i$ and $A_i + B_i K_i$ is stable; Γ_i and Π_i satisfy (4.1) for each (A_i, B_i, C_i, P_i, S, Q_i), $i = 1...r$.

The output of the fuzzy controller is given by

$$u(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))[K_i x(t) + L_i \omega(t)]}{\sum_{i=1}^{r} \lambda_i(z(t))} \tag{4.5}$$

The design purpose in this study is to specify the fuzzy control in (4.5) to achieve a global stable and optimal output control. The following result is then obtained:

Theorem: If the local linear system in (4.3) satisfies Assumption (A1), the local output regulator (4.4) can asymptotically stabilize the local linear system in (4.3) and ensure the asymptotic output tracking for every possible initial state and every possible exogeneous input. Then, the fuzzy control (4.5) can globally stabilize the Takagi-Sugeno system (4.3) and ensure the asymptotic output tracking. Furthermore, its output is also optimal.

Proof: If connecting plant rules with controller rules, a closed loop system is obtained:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))A_i x(t)}{\sum_{i=1}^{r} \lambda_i(z(t))} + \frac{\sum_{i=1}^{r} \lambda_i(z(t))B_i}{\sum_{i=1}^{r} \lambda_i(z(t))}\left(\frac{\sum_{i=1}^{r} \lambda_i(z(t))(K_i x(t) + L_i \omega(t))}{\sum_{i=1}^{r} \lambda_i(z(t))}\right) + \frac{\sum_{i=1}^{r} \lambda_i(z(t))P_i \omega(t)}{\sum_{i=1}^{r} \lambda_i(z(t))}$$

$$\dot{\omega}(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))S\omega(t)}{\sum_{i=1}^{r} \lambda_i(z(t))}$$

First let $\omega(t) = 0$ and prove that the equilibrium x=0 is globally asymptotically stable in the first approximation. Denote the sum over all possible combinations of $\left(\sum_{i=1}^{r} \lambda_i(z(t))\right)\left(\sum_{i=1}^{r} \lambda_i(z(t))\right)$ as $\sum_{i,j}$

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))A_i x(t)}{\sum_{i=1}^{r} \lambda_i(z(t))} + \frac{\sum_{i=1}^{r} \lambda_i(z(t))B_i}{\sum_{i=1}^{r} \lambda_i(z(t))}\left(\frac{\sum_{i=1}^{r} \lambda_i(z(t))(K_i x(t))}{\sum_{i=1}^{r} \lambda_i(z(t))}\right)$$

$$=\frac{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))(A_i+B_iK_j)}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}x(t)$$

For stability analysis, a Lyapunov function is chosen:

$$V(x)=x^T P_l x$$

where P_l is a symmetric positive definite matrix. The derivative of $V(x)$ can be written as:

$$\dot{V}(x)=x^T P_l\dot{x}+\dot{x}^T P_l x$$

$$\begin{aligned}\dot{V}(x)&=x^T P_l\dot{x}+\dot{x}^T P_l x\\ &=x^T P_l\left(\frac{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))(A_i+B_iK_j)}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}\right)x+x^T\left(\frac{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))(A_i+B_iK_j)}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}\right)^T P_l x\\ &=x^T\left(\begin{array}{l}P_l\left(\frac{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))(A_i+B_iK_j)}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}\right)+\\ \left(\frac{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))(A_i+B_iK_j)}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}\right)^T P_l\end{array}\right)x\\ &=x^T\left(\frac{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))\begin{pmatrix}P_l(A_i+B_iK_j)+\\(A_i+B_iK_j)^T P_l\end{pmatrix}}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}\right)x\end{aligned}$$

since

$$0\le\frac{\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))}\le 1$$

we have

$$\begin{aligned}\dot{V}(x)&\le\sum_{i,j=1}^{r}x^T\\ &\quad\left(P_l(A_i+B_iK_j)+(A_i+B_iK_j)^T P_l\right)x\\ &\le x^T P_l\\ &\quad\left(\sum_{i=1}^{r}(A_i+B_iK_j)+\left(\sum_{i=1}^{r}(A_i+B_iK_j)\right)^T\right)\\ &\quad P_l x\end{aligned}$$

Because $A_i + B_i K_i$ is stable *i.e.*, $A_i + B_i K_i < 0$.

$\sum_{i=1}^{r}(A_i + B_i K_j) < 0$

linear matrix inequality : $P_l \sum_{i=1}^{r}(A_i + B_i K_j) + (\sum_{i=1}^{r}(A_i + B_i K_j))^T P_l < 0$ is therefore established.

$\dot{V}(x) < 0$

The equilibrium x=0 is globally asymptotically stable in the first approximation.

Secondly, it should be demonstrated that the output of the system is optimal when $\omega(t) = 0$. Consider the quadratic cost function for the whole system:

$$J = \int_0^{\infty} (e^T \Lambda e + u^T Ru) dt$$

According to the "additive property of energy" [27] the whole cost J is the combination of the local cost: J_i.

$$J_i = \int_0^{\infty} (e^T \Lambda e + u_i^T Ru_i)$$

where the subscript i can be any element of set $\{1,\ldots,r\}$, which means any ruler of the fuzzy system. Now, based on this additive property of energy, we know that, at any time instant, because we can find the optimal control law for each subsystem to minimize the local cost, J_i, the fuzzy "blended" control law (4.4) is the global minimizer of the total cost, J, because the optimal control law for each subsystem can be found to minimize the local cost, J_i.

When $\omega(t) \neq 0$, it can be shown that $\lim_{t \to \infty} e(t) = 0$.

$$\begin{aligned} e(t) &= \frac{\sum_{i=1}^{r} \lambda_i(z(t)) C_i x(t)}{\sum_{i=1}^{r} \lambda_i(z(t))} + \frac{\sum_{i=1}^{r} \lambda_i(z(t)) Q_i \omega(t)}{\sum_{i=1}^{r} \lambda_i(z(t))} \\ &= \frac{\sum_{i=1}^{r} \lambda_i(z(t))(C_i x(t) + Q_i \omega(t))}{\sum_{i=1}^{r} \lambda_i(z(t))} \end{aligned}$$

Because $0 \le \frac{\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))} \le 1$, $0 \le e(t) \le \sum_{i=1}^{r}(C_i x(t) + Q_i\omega(t)) = \sum_{i=1}^{r}(e_i(t))$. Because the equation $\Pi S = A\Pi + B\Gamma + P$ is satisfied, the graph of the mapping $x = \Pi\omega$ is a center manifold of the Linear System [28]. By the equation ($0 = C\Pi + Q$), we obtain:

$$
\begin{aligned}
e(t) &= Cx(t) + Q\omega(t) - (C\Pi + Q)\omega(t) \\
&= Cx(t) - C\Pi\omega(t)
\end{aligned}
$$

The point $(x,\omega) = (0,0)$ is a stable equilibrium of the Linear System. Then, for sufficiently small $(x(0),\omega(0))$, the solution ($x(t),\omega(t)$) of the Linear System remains in any arbitrarily small neighborhood of (0,0) for all t ≥ 0. Using a property of center manifold [6], it is deduced that there exist real numbers M>0 and a>0 such that

$$\|x(t) - \Pi(\omega(t))\| \le Me^{-at}\|x(0) - \Pi(\omega(0))\|$$

for all t>0. By continuity of e(t), $\lim_{t\to\infty} e_i(t) = 0$. So $\lim_{t\to\infty}\sum_{i=1}^{r} e_i(t) = 0$. And because $\lim_{t\to\infty} 0 = 0$,and $\lim_{t\to\infty} e(t) = 0$, so there exists a neighborhood of (0,0), for each initial state and each possible exogenous input $(x(0),\omega(0))$, $\lim_{t\to\infty} e(t) = 0$.

Chapter 5

Experiment Setup

5.1. Linear Model

To demonstrate the validity of the proposed fuzzy control law, a real-time implementation of the control strategy was developed for the Pendubot. As demonstrated in chapter 3, the state equation for the Pendubot given in (2.9) is known to be:

$$\dot{x}(t) = f(x) + g(x)u(t) = F(x(t), u(t), t)$$

where $x = [q_1 \quad \dot{q}_1 \quad q_2 \quad \dot{q}_2]^T$ is the state vector and $u(t) = [\tau_1 \quad 0]^T$ is the input vector. The output is selected as y(t)=q_2. A linear model for a specific equilibrium point can be obtained using the Taylor series. For the Pendubot every equilibrium point must satisfy: $q_2 + q_1 = 90^0$. As in chapter 2, the linearization of equation by the Taylor series is given as:

$$\delta\dot{x} = F_x(t)\delta x + F_u(t)\delta u$$

where

$$F_x(t) = \frac{\partial F(x(t), u(t), t)}{\partial x}\Big|x^0, u^0$$

$$F_u(t) = \frac{\partial F(x(t), u(t), t)}{\partial u}\Big|x^0, u^0$$

$x^0 = [q_1^0, \dot{q}_1^0, q_2^0, \dot{q}_2^0]^T$; and $u^0 = [\tau_1^0 \quad 0]^T$ are the values of x and u for the specific equilibrium point.

Defining $x := \delta x$; $A := F_x(t)$; $B := F_u(t)$, we have

$$\dot{x} = Ax(t) + Bu(t)$$

5.2. Experimental Application

In this section the real time implementation of the proposed algorithm to the Pendubot is discussed. The objective is to force the angular position of link 2 (q_2) to follow three sinusoidal signals. The amplitude of the signals are 50^0, 60^0, 70^0. The experimental results will be compared and conclusions made. In order to track this signal, it is required that at every trajectory point the equality $q_1 + q_2 = 90^0$ has to be filled.

5.2.1. Trajectory Tracking around an unstable top equilibrium point

Fuzzy plant: To model the Pendubot, we propose the five fuzzy sets in Fig. 5.1. The notations are BP (Big Positive), MP (Medium Positive), Z (Zero), MN (Medium Negative), BN (Big Negative). To obtain the linear models for the consequents, the nonlinear model of the Pendubot is linearized around the following equilibrium points

$$x_i^0 = \left[q_{1i}^0, 0, q_{2i}^0, 0\right],\ i = 1,\dots,5$$

where $q_{21}^0 = 70^0$, $q_{22}^0 = 35^0$, $q_{23}^0 = 0^0$, $q_{24}^0 = -35^0$, $q_{25}^0 = -70^0$ and $q_{1i}^0 = 90^0 - q_{2i}^0$, $i = 1,\dots,5$

For each equilibrium point we obtain a linear system (A_i, B_i), $i = 1,\dots,5$. The output y is always chosen as q_2, so C=[0 0 1 0] is valid for all linear systems. The system has no external perturbation, hence the matrix P=0.

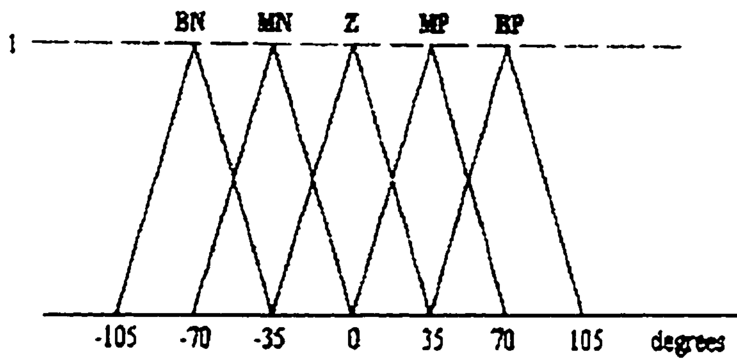

Fig. 5.1 Fuzzy Sets

The control goal is to track a sinusoidal signal obtained by adding an additional signal (offset) to the sinusoidal signal for each equilibrium point: hence the following reference has to be generated.

$$y_r = u_{ofsi} + k \sin \alpha t; \qquad i = 1.....5$$

where u_{ofsi} is the required offset for each equilibrium point and is equal to q_{2i}^{0}. k is the amplitude of the sinusoidal signal and α is its angular frequency.

The exosystem for each linear region is given as:

$$\dot{\omega}(t) = S\omega(t); \qquad \omega(0) = \omega_0$$
$$y_r(t) = Q_i\omega(t)$$

where:

$$\omega^T = [\omega_1(t) \quad \omega_2(t) \quad \omega_3(t)]$$

$$\omega_0^{\,T} = [1 \quad 0 \quad 1]$$

$$S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \alpha \\ 0 & -\alpha & 0 \end{bmatrix} \qquad \forall i = 1.....5$$

$$Q_i = [-u_{ofsi} \quad -k \quad 0] \qquad i = 1.....5$$

$\omega_1(t)=1$; $\omega_2(t)=\sin 0.5t$; $\omega_3(t)=\cos 0.5t$;

Fuzzy controller: The fuzzy sets for the controller are the same as for the plants.

For each $(A_i, B_i, C_i, P_i, S, Q_i)$ the matrices Γ_i and Π_i are determined from (4.1). We use "maple" to calculate them (Appendix B). The L_i is calculated using (4.2). each K_i is selected according to the linear optimal control theory. We use "Matlab" to calculate them (Appendix C). The local control signals are obtained from (4.3). The particular values of K and L for each local controller are given in (appendix D). Finally the control signal applied to the system is given by (4.4)

Fuzzy rules: There are five rules for the plant and for the controller.

1^{st} Plant Rule

If q_2 is big positive

$$\text{Then}\begin{cases}\dot{x}(t)=A_1x(t)+B_1u(t)\\ \dot{\omega}(t)=S\omega(t)\\ e(t)=Cx(t)+Q_1\omega(t)\end{cases}$$

1^{st} Controller Rule

If $q_2(t)$ is big positive

Then $u=K_1x(t)+L_1\omega(t)$

2^{nd} Plant Rule

If q_2 is medium positive

$$\text{Then}\begin{cases}\dot{x}(t)=A_2x(t)+B_2u(t)\\ \dot{\omega}(t)=S\omega(t)\\ e(t)=Cx(t)+Q_2\omega(t)\end{cases}$$

2^{nd} Controller Rule

If $q_2(t)$ is medium positive.

Then $u = K_2 x(t) + L_2 \omega(t)$

3^{rd} Plant Rule

IF $q_2(t)$ is zero.

$$\text{Then}\begin{cases} \dot{x}(t) = A_3 x(t) + B_3 u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_3 \omega(t) \end{cases}$$

3^{rd} Controller Rule

If $q_2(t)$ is zero.

Then $u = K_3 x(t) + L_3 \omega(t)$

4^{th} Plant Rule

If q_2 is medium negative

$$\text{Then}\begin{cases} \dot{x}(t) = A_4 x(t) + B_4 u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_4 \omega(t) \end{cases}$$

4^{th} Controller Rule

If $q_2(t)$ is medium negative.

Then $u = K_4 x(t) + L_4 \omega(t)$

5^{th} Plant Rule

If q_2 is big negative

$$\text{Then}\begin{cases} \dot{x}(t) = A_5 x(t) + B_5 u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_5 \omega(t) \end{cases}$$

5^{th} Controller Rule

If $q_2(t)$ is big negative.

Then $u = K_s x(t) + L_s \omega(t)$

5.2.2. Trajectory Tracking around an unstable mid equilibrium point

To demonstrate the advantages of our fuzzy algorithm, the Pendubot is forced to track three desired trajectories around the middle equilibrium point. The amplitude of the signals are 50^0, 60^0, and 70^0. It is the counterpart of section 5.2.1.

Fuzzy plant: to model the Pendubot, the five fuzzy sets in Fig. 5.2 are proposed. The notations are BL (Big Larger PI), ML (Medium Larger PI), E (Equal to PI), MS (Medium Smaller PI), BS (Big Smaller PI). To obtain the linear models for the consequents, the nonlinear model of the Pendubot is linearized around the following equilibrium points:

$x_i^0 = \left[q_{1i}^0, 0, q_{2i}^0, 0\right]$, $i = 1,...,5$

where $q_{21}^0 = -160^0$, $q_{22}^0 = -125^0$, $q_{23}^0 = -90^0$, $q_{24}^0 = -55^0$, $q_{25}^0 = -20^0$ and $q_{1i}^0 = 90^0 - q_{2i}^0$, $i = 1,...,5$

For each equilibrium point we obtain a linear system (A_{mi}, B_{mi}), $i = 1,....,5$. The values of (A_{mi}, B_{mi}) are giving in appendix D. The output y is always chosen as q_2, so C=[0 0 1 0] is valid for all linear systems. The system has not external perturbation, hence the matrix P=0.

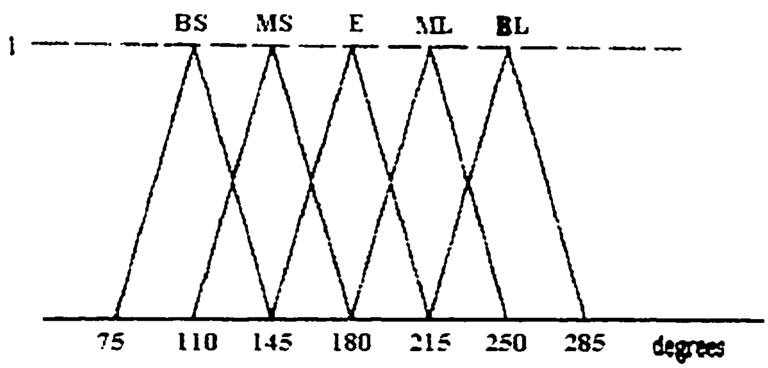

Fig 5 2 Fuzzy Sets

The reference signal and the exosystem are similar to those in section 5.2.1 except the offset. It is equal to $\pi - q_{2s}^{0}$.

Fuzzy controller: The design procedure of the control signal applied to the system is the same in section 5.2.1.

1^{st} Plant Rule

IF q_2 is big smaller PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_{m1}x(t) + B_{m1}u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_{m1}\omega(t) \end{cases}$$

1^{st} Controller Rule

IF $q_2(t)$ is big smaller PI.

Then $u = K_{m1}x(t) + L_{m1}\omega(t)$

2^{nd} Plant Rule

IF q_2 is medium smaller PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_2x(t) + B_2u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_2\omega(t) \end{cases}$$

2^{nd} Controller Rule

IF $q_2(t)$ is medium smaller PI.

Then $u = K_{m2}x(t) + L_{m2}\omega(t)$

3^{rd} Plant Rule

IF q_2 is equal to PI

$$\text{Then}\begin{cases}\dot{x}(t)=A_{m3}x(t)+B_{m3}u(t)\\ \dot{\omega}(t)=S\omega(t)\\ e(t)=Cx(t)+Q_{m3}\omega(t)\end{cases}$$

3rd Controller Rule

IF $q_2(t)$ is equal to PI.

Then $u=K_{m3}x(t)+L_{m3}\omega(t)$

4th Plant Rule

IF q_2 is medium larger PI

$$\text{Then}\begin{cases}\dot{x}(t)=A_{m4}x(t)+B_{m4}u(t)\\ \dot{\omega}(t)=S\omega(t)\\ e(t)=Cx(t)+Q_{m4}\omega(t)\end{cases}$$

4th Controller Rule

IF $q_2(t)$ is medium larger PI.

Then $u=K_{m4}x(t)+L_{m4}\omega(t)$

5th Plant Rule

IF q_2 is big larger PI

$$\text{Then}\begin{cases}\dot{x}(t)=A_{m5}x(t)+B_{m5}u(t)\\ \dot{\omega}(t)=S\omega(t)\\ e(t)=Cx(t)+Q_{m5}\omega(t)\end{cases}$$

5th Controller Rule

IF $q_2(t)$ is big larger PI.

Then $u=K_{m5}x(t)+L_{m5}\omega(t)$

5.3. Experiments Results

The schematic diagram of our experimental setup is illustrated in section 2.1. Our control algorithm was implemented in C language on a Pentium IBM PC 125 MHZ. The sampling time is 5ms. The code is in Appendix (E. F). Fig. 5.3-Fig. 5.6 are the experimental results of the trajectory tracking around the top equilibrium point. Fig.5.3 shows the results when the sinusoidal signal tracked is 50 sin0.5t. Fig. 5.3(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 and the dotted line is the desired trajectory to be followed. Fig. 5.3(b) gives the trajectory of q_1. Fig. 5.3(c) is the output tracking error and the Fig. 5.3(d) is the control signal.

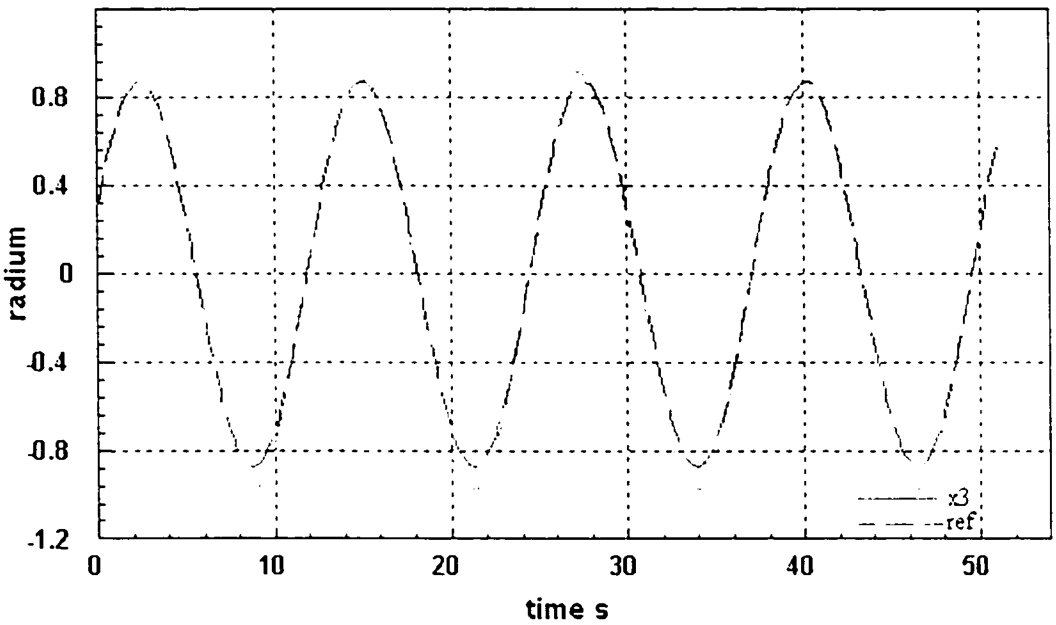

Fig. 5.3(a) Link 2 Angular Position

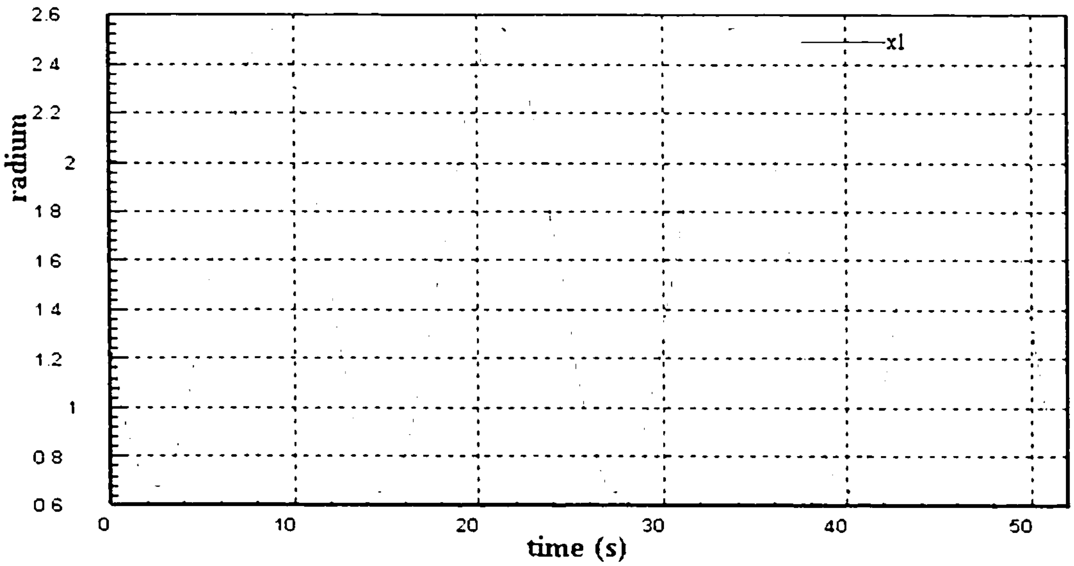

Fig. 5.3(b) Link 1 Angular Position

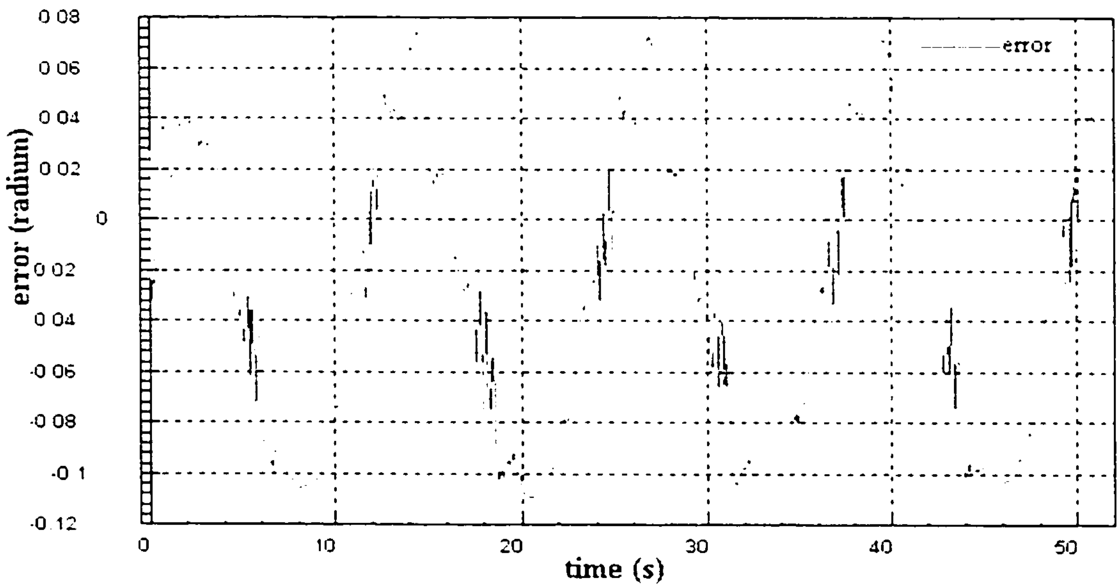

Fig. 5.3(c) Tracking Error

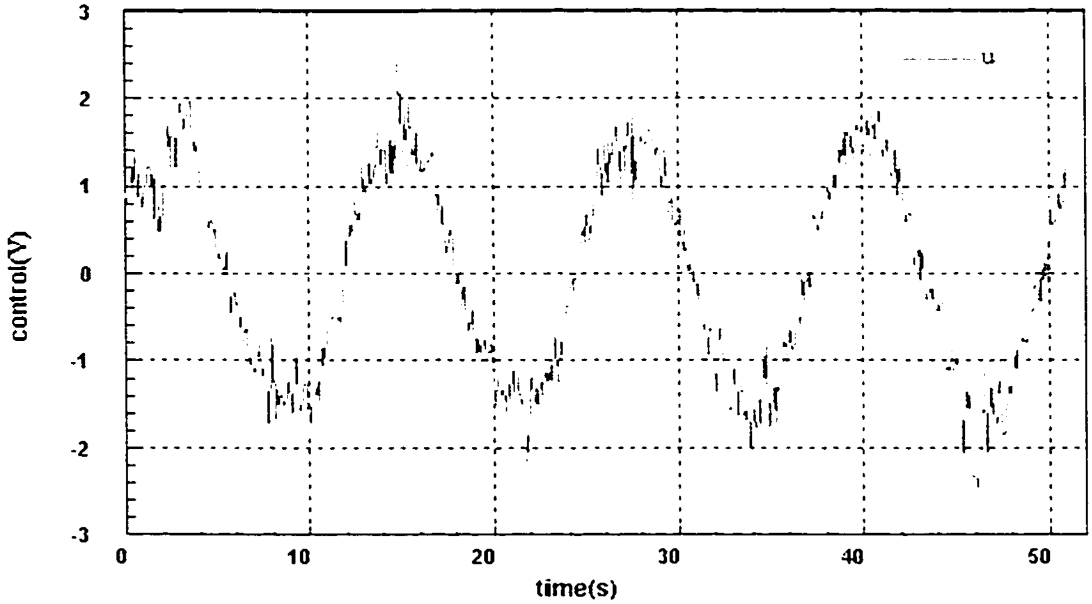

Fig. 5.3(d) Control Signal

Fig. 5.4 shows the results when the tracked sinusoidal signal is 60 sin0.5t. Fig. 5.4(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 and the dotted line is the desired trajectory to be followed. Fig. 5.4(b) gives the trajectory of q_1. Fig. 5.4(c) is the output tracking error and the Fig. 5.4(d) is the control signal.

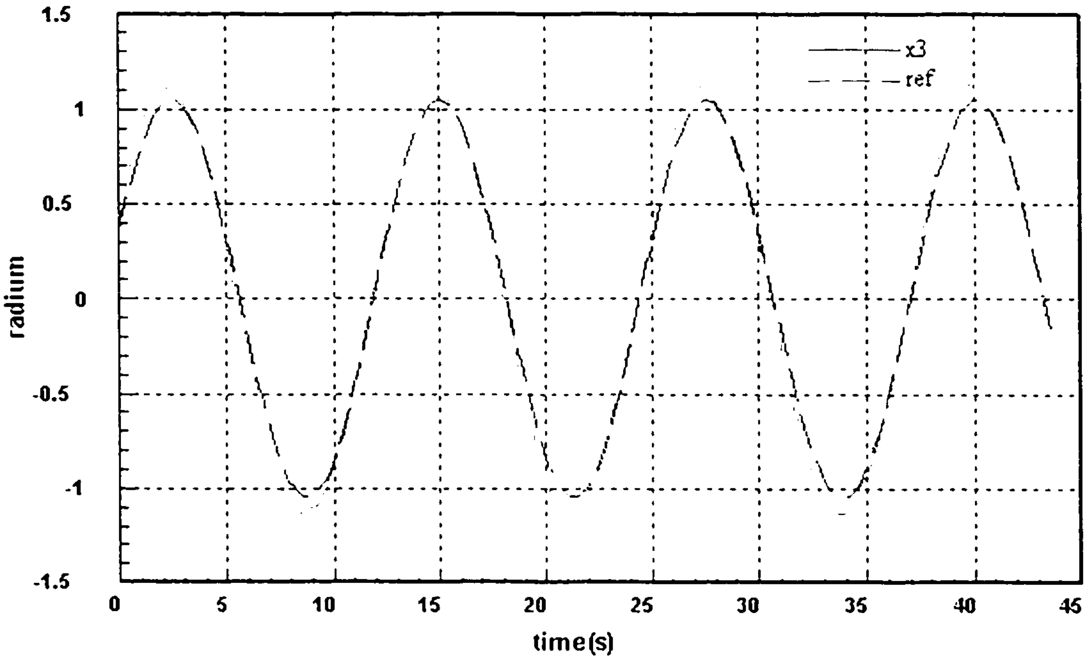

Fig. 5.4(a) Link 2 Angular Position

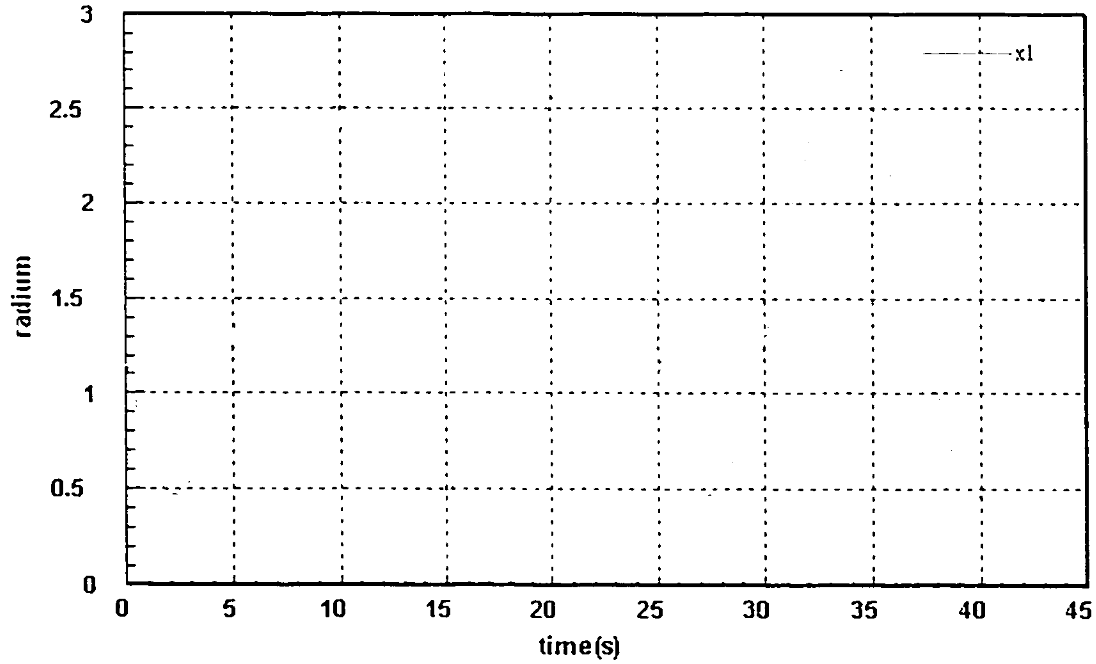

Fig. 5.4(b) Link 1 Angular Position

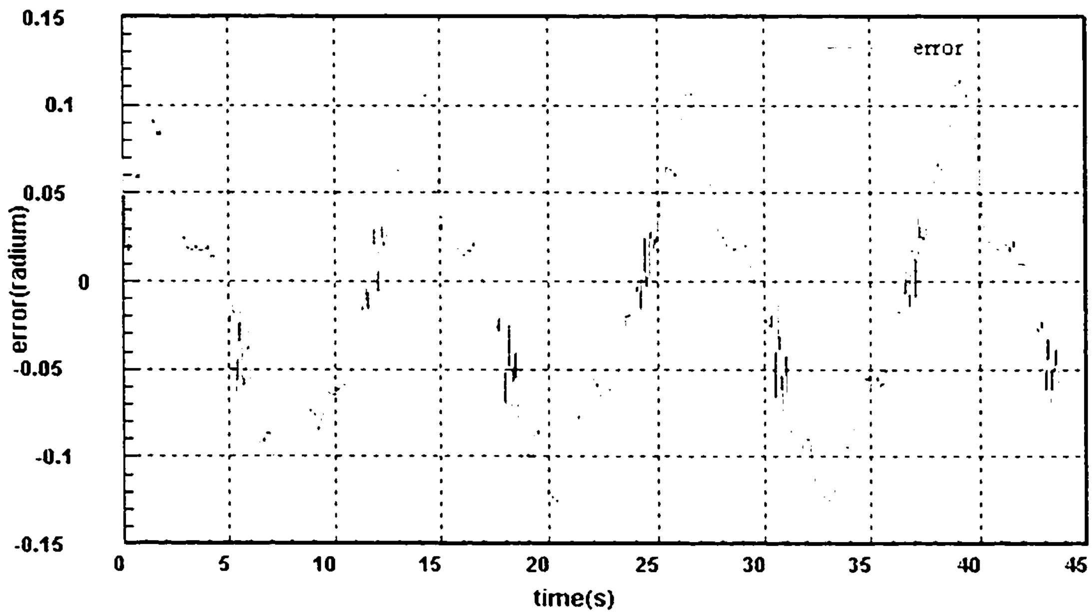

Fig. 5.4(c) Tracking Error

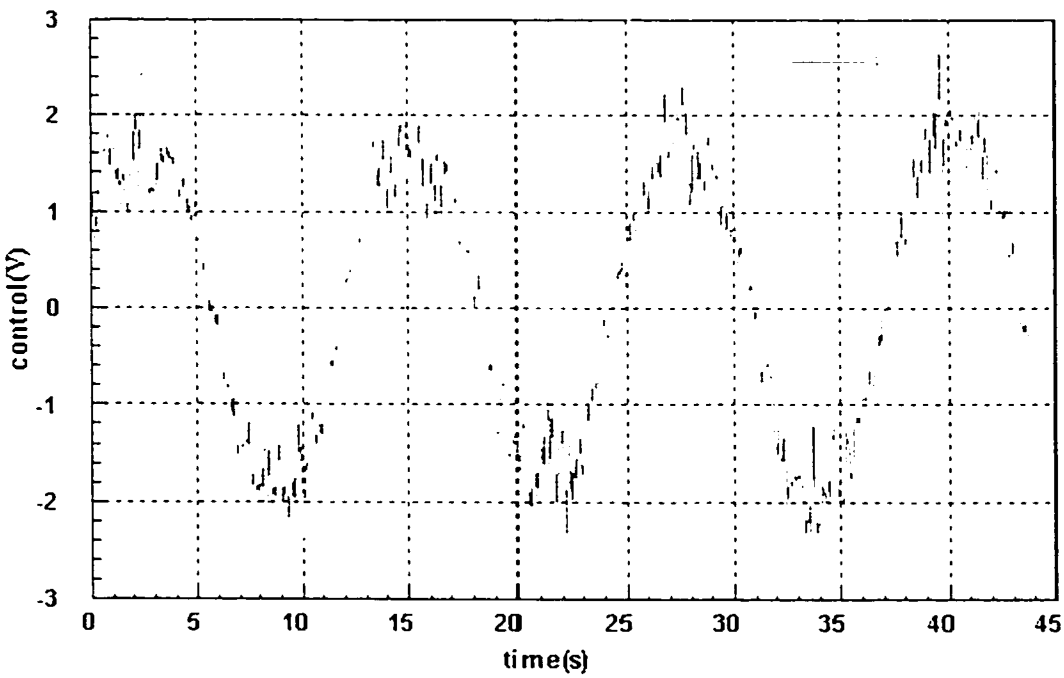

Fig. 5.4(d) Control Signal

Fig. 5.5 shows results when the sinusoidal signal tracked is 70 sin0.5t. Fig. 5.5(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 while the dotted line is the desired trajectory to be followed. Fig. 5.5(b) gives the trajectory of q_1. Fig. 5.5(c) is the output tracking error and the Fig. 5.5(d) is the control signal.

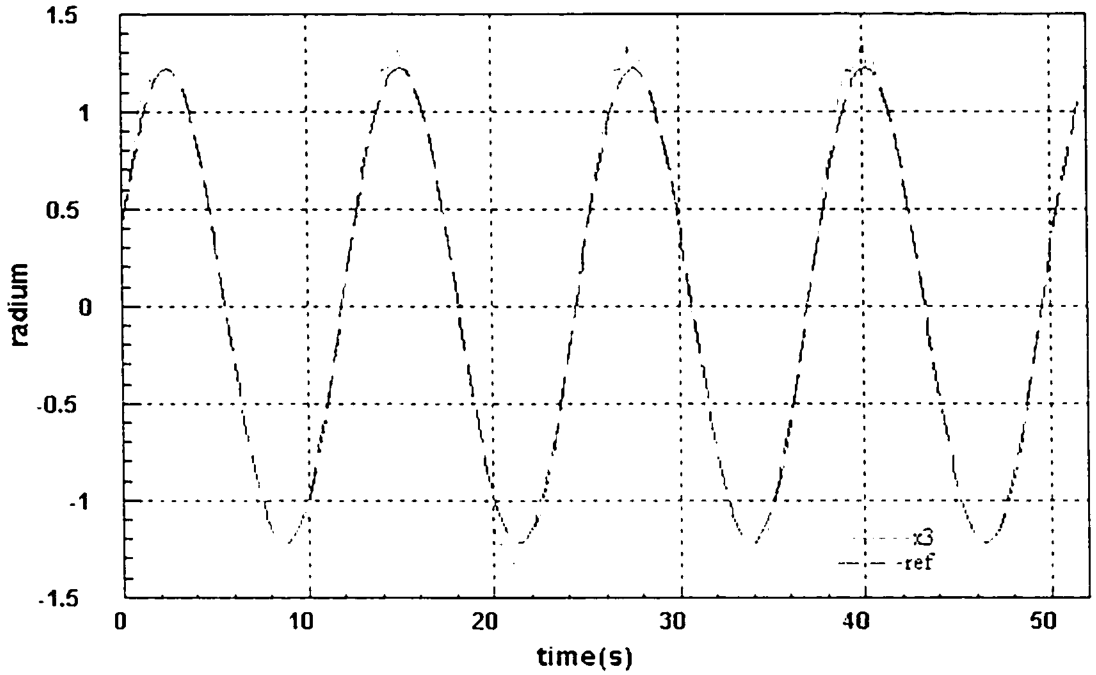

Fig. 5.5(a) Link 2 Angular Position

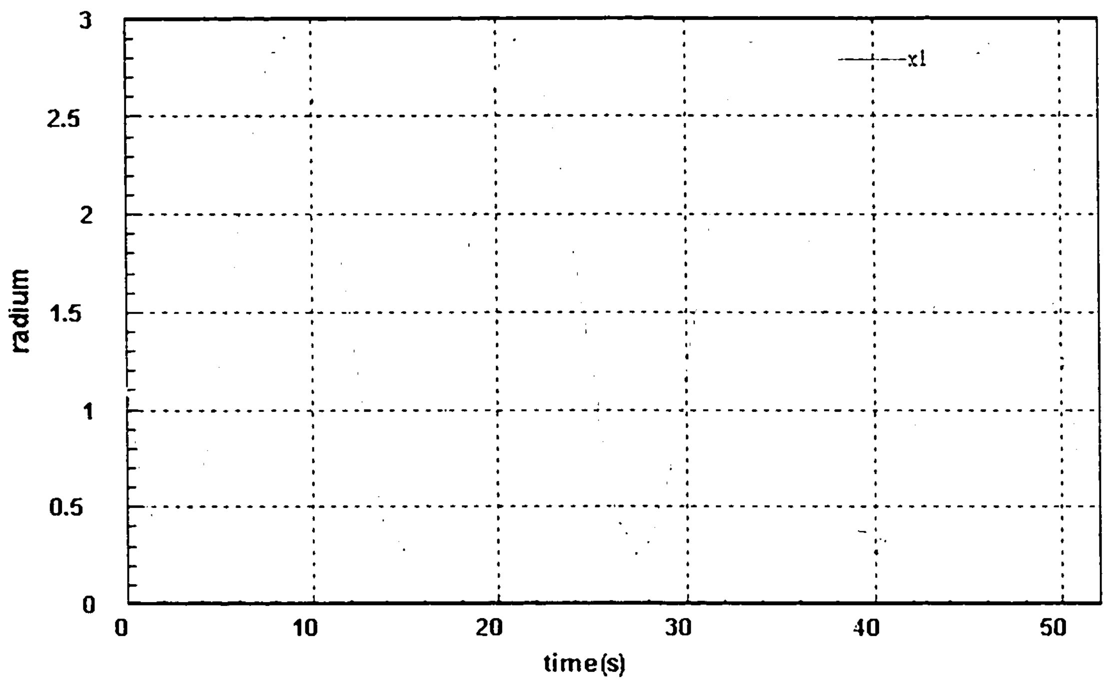

Fig. 5.5(b) Link 1 Angular Position

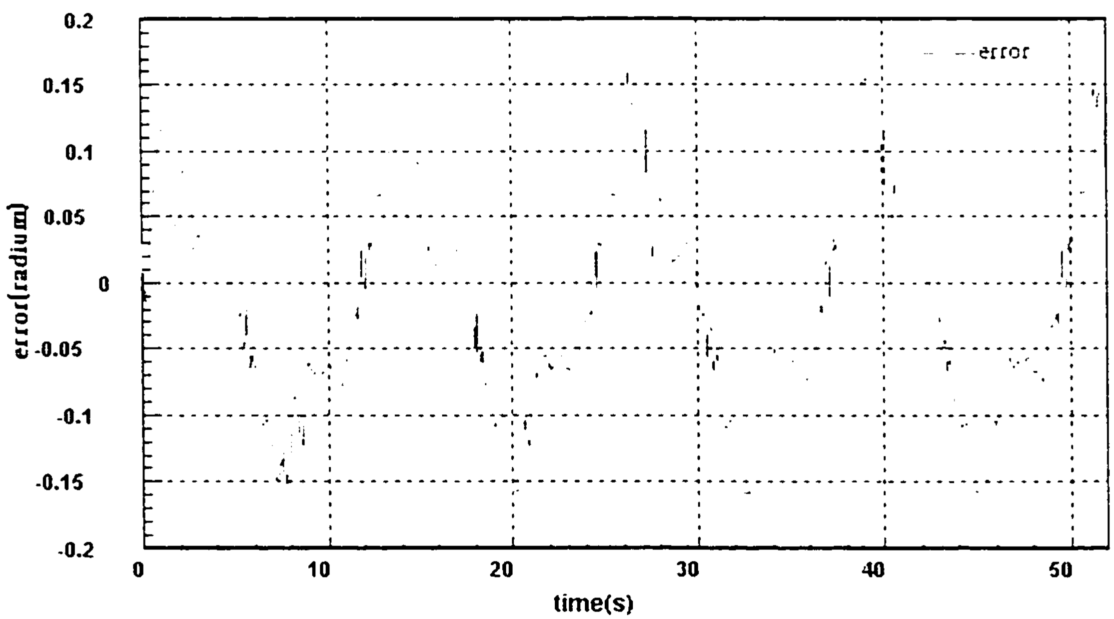

Fig. 5.5(c) tracking error

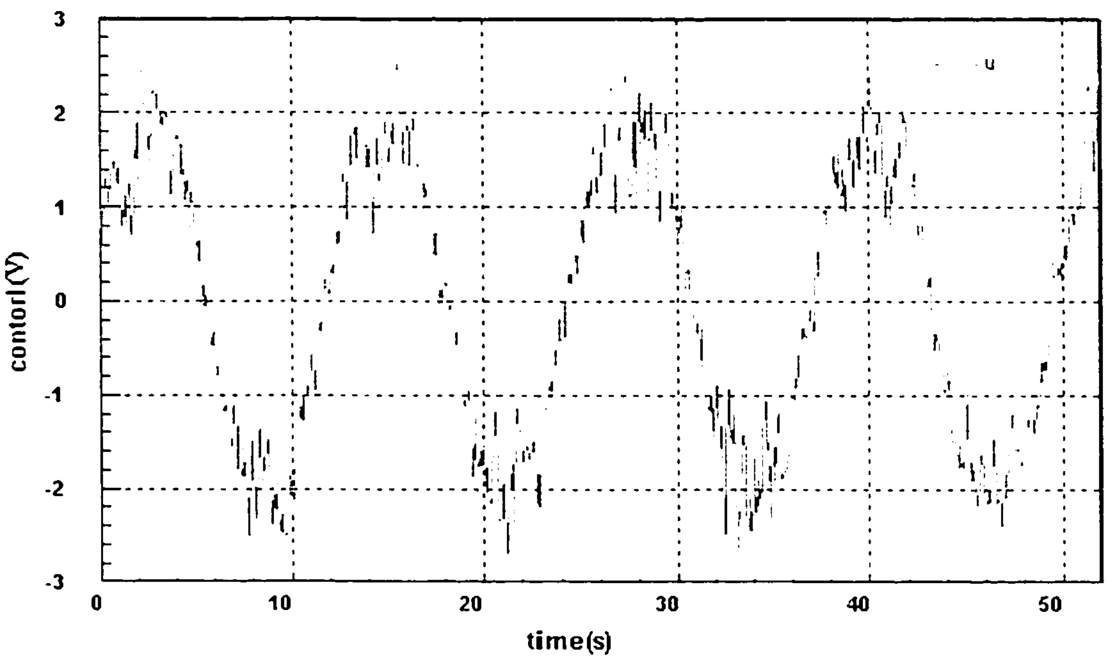

Fig. 5.5(d) Control Signal

Fig. 5.6 (a)-Fig. 5.6(d) are some photos illustrating the real-time trajectory tracking around the top equilibrium point.

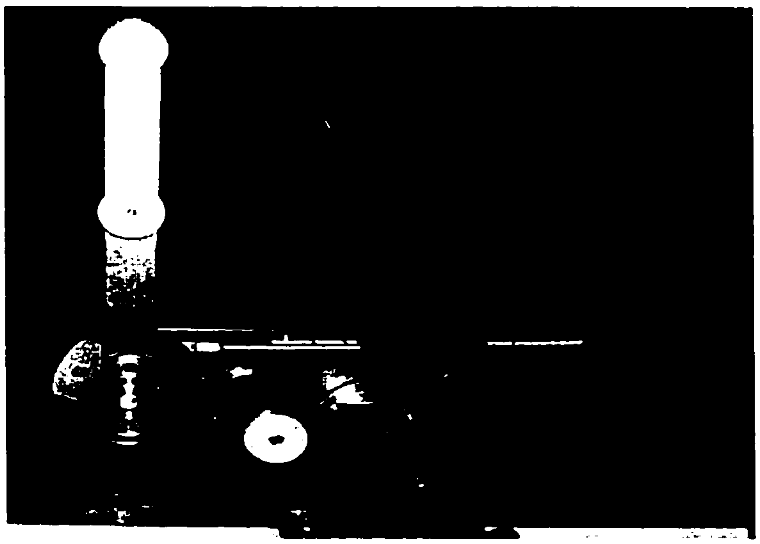

Fig. 5.6(a) Trajectory Tracking around the Top Equilibrium Point

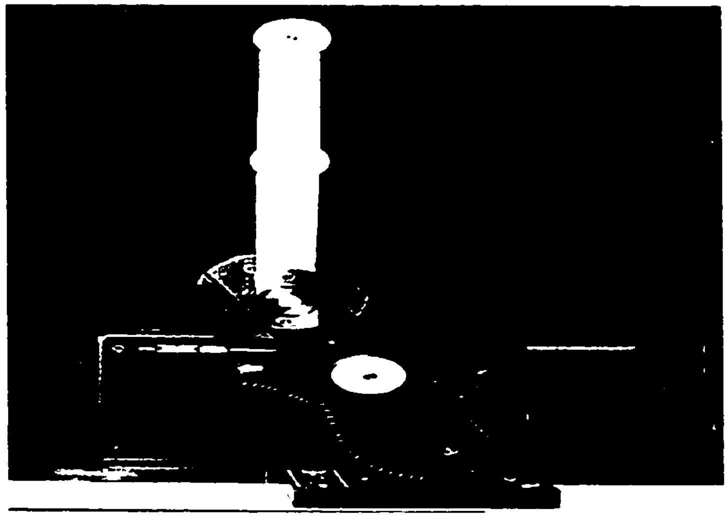

Fig. 5.6 (b) Trajectory Tracking around the Top Equilibrium Point

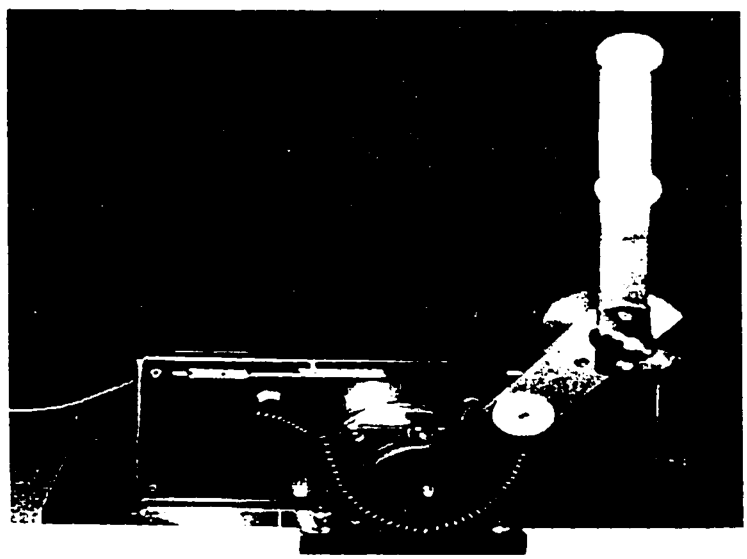

Fig. 5.6 (c) Trajectory Tracking around the Top Equilibrium Point

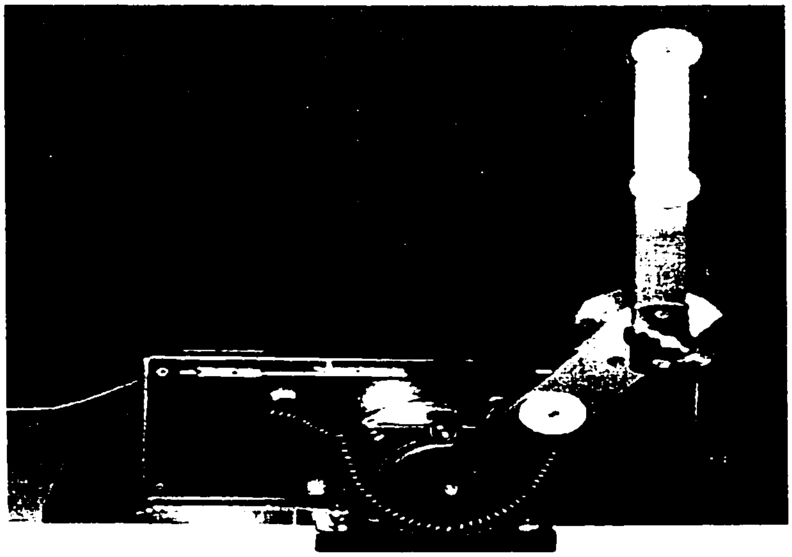

Fig. 5.6 (d) Trajectory tracking around the Top Equilibrium Point

Fig. 5.7-5.9 are the experimental results of the trajectory tracking around the mid equilibrium point. Fig. 5.7 shows the results when the tracked sinusoidal signal is 180+50 sin0.5t. Fig. 5.7(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 and the dotted line is the desired trajectory to be followed. Fig. 5.7(b) gives the trajectory of q_1. Fig. 5.7(c) is the output tracking error and the Fig. 5.7(d) is the control signal.

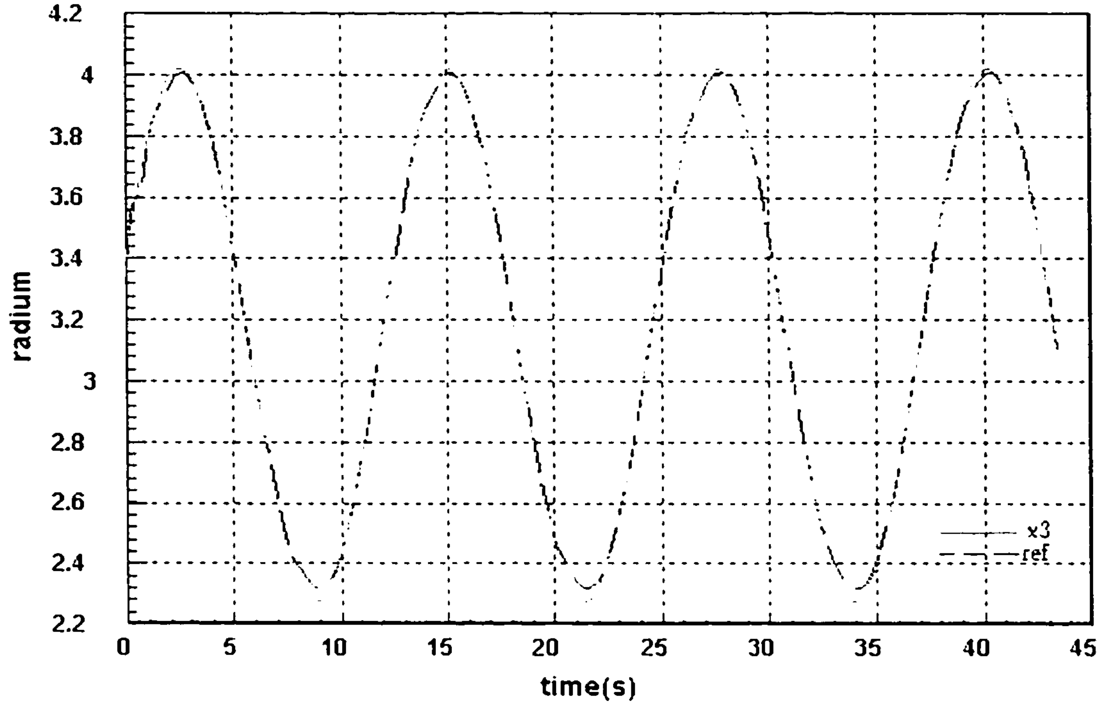

Fig. 5.7(a) Link 2 Angular Position

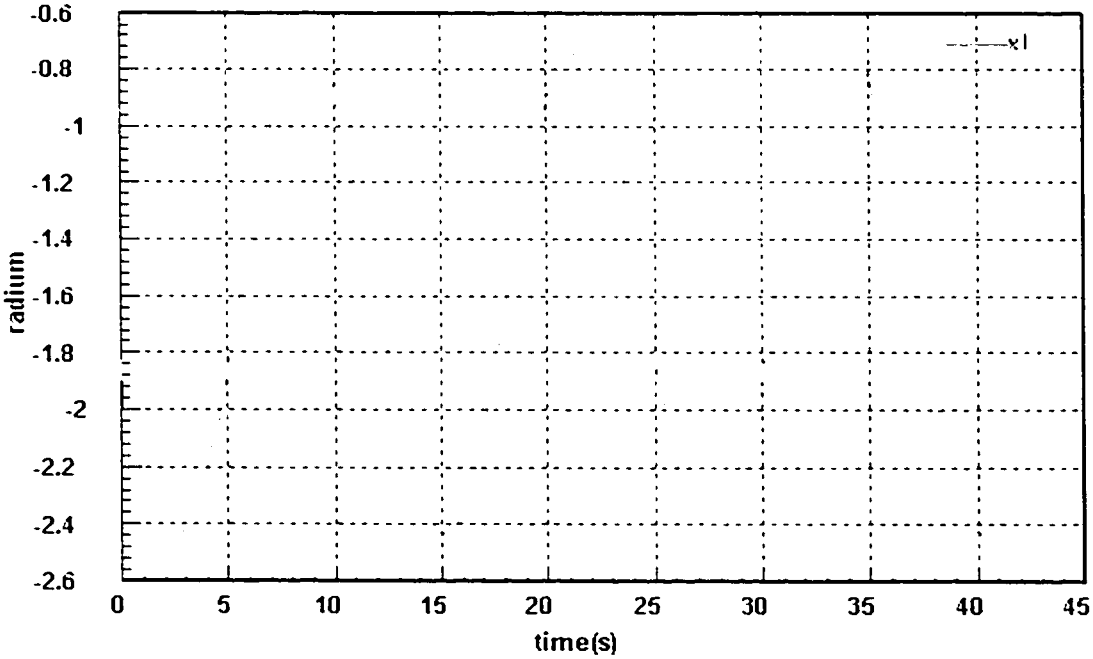

Fig. 5.7(b) Link 1 Angular Position

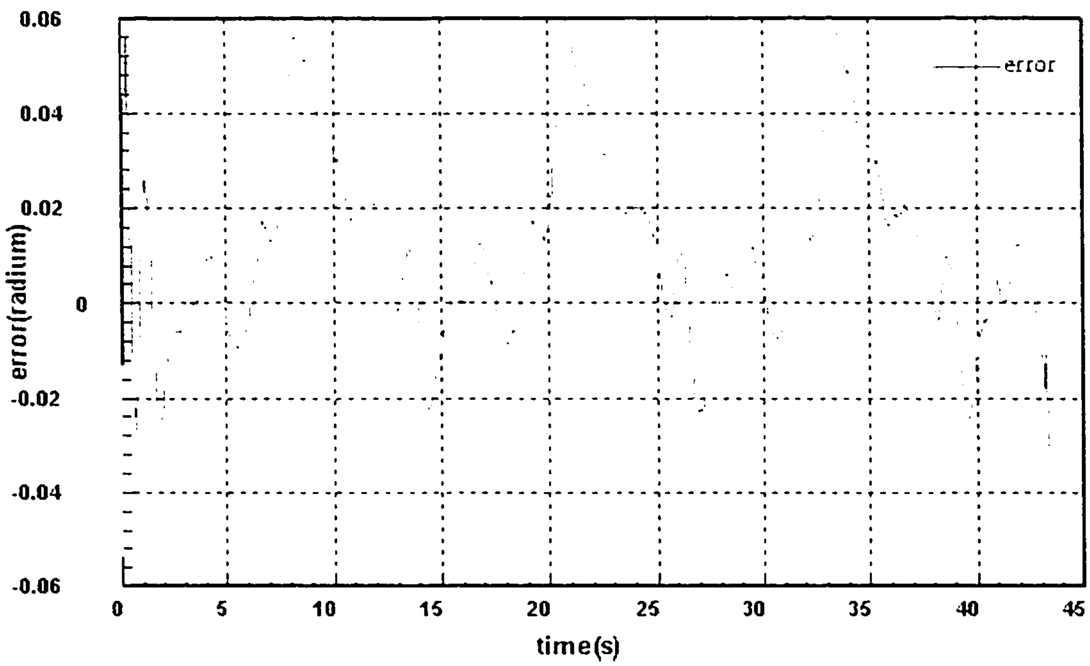

Fig. 5.7(c) Tracking Error

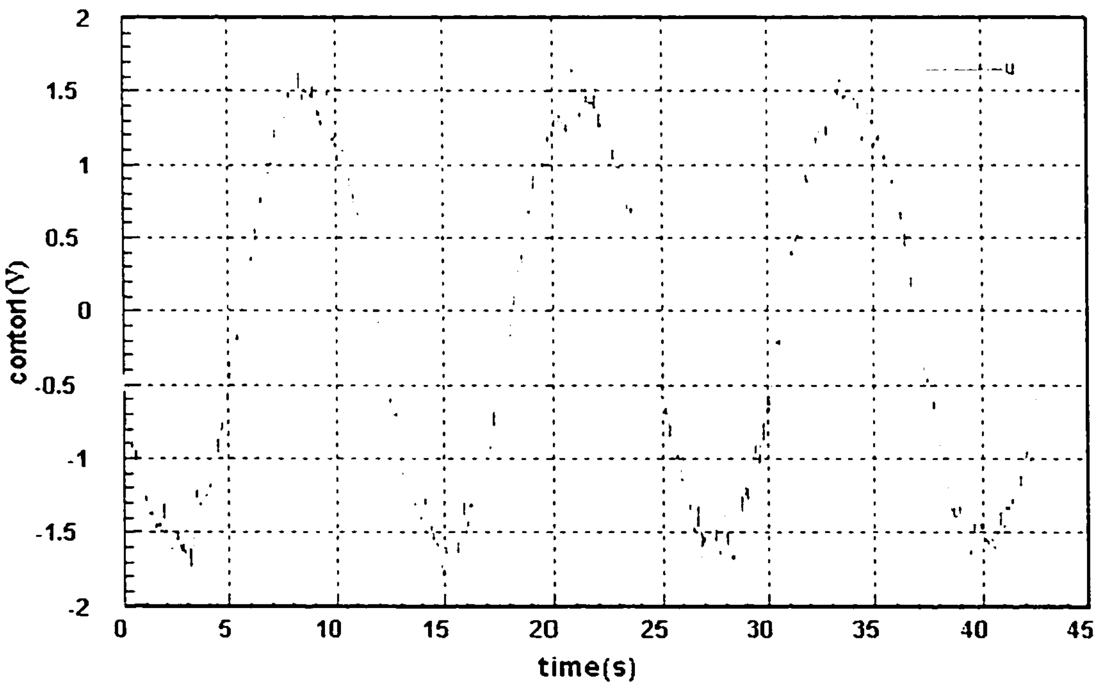

Fig. 5.7(d) Control Signal

Fig. 5.8 shows the results when the tracked sinusoidal signal is 180+60 sin0.5t. Fig. 5.8(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 and the dotted line is the desired trajectory to be followed. Fig. 5.8(b) gives the trajectory of q_1. Fig. 5.8(c) is the output tracking error and the Fig. 5.8(d) is the control signal.

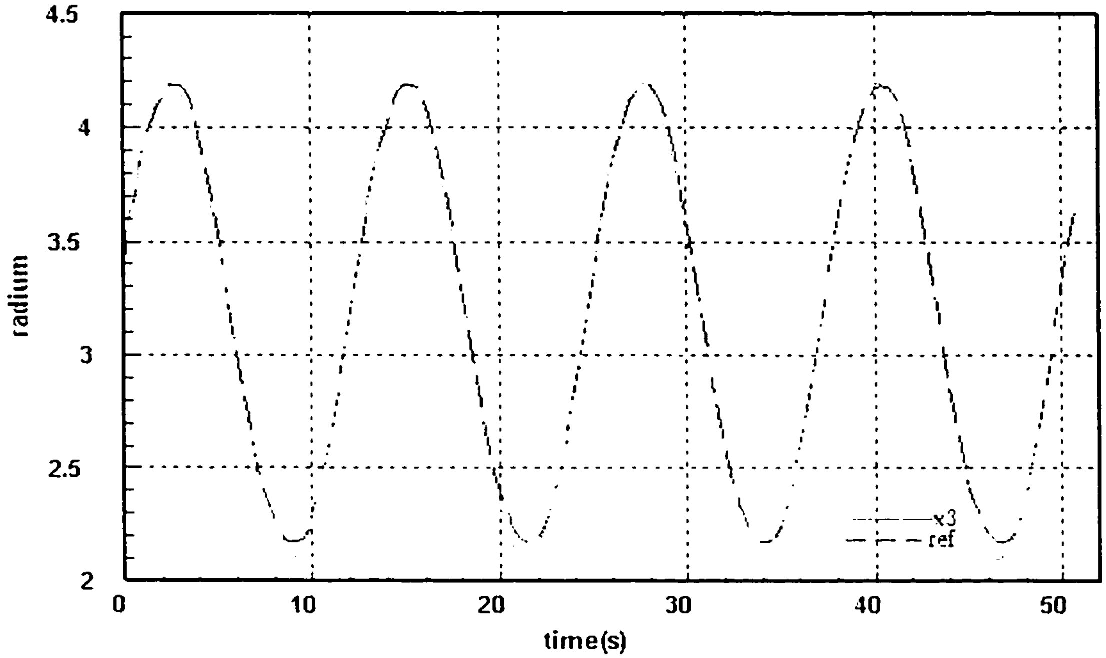

Fig. 5.8(a) Link 2 Angular Position

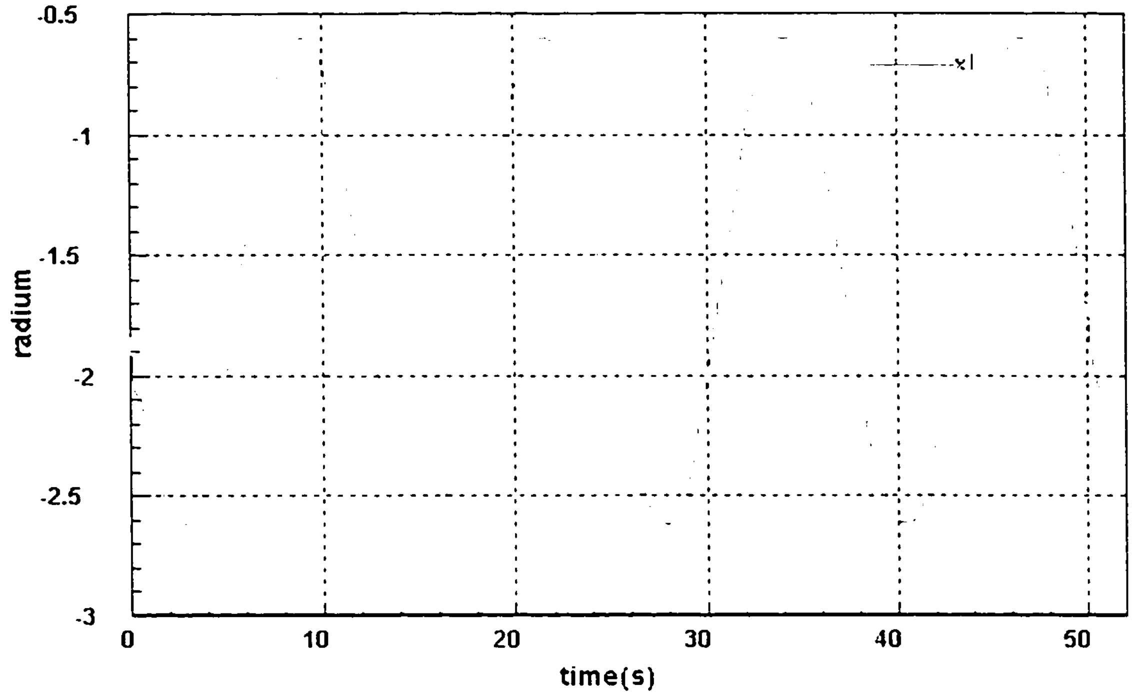

Fig. 5.8(b) Link 1 Angular Position

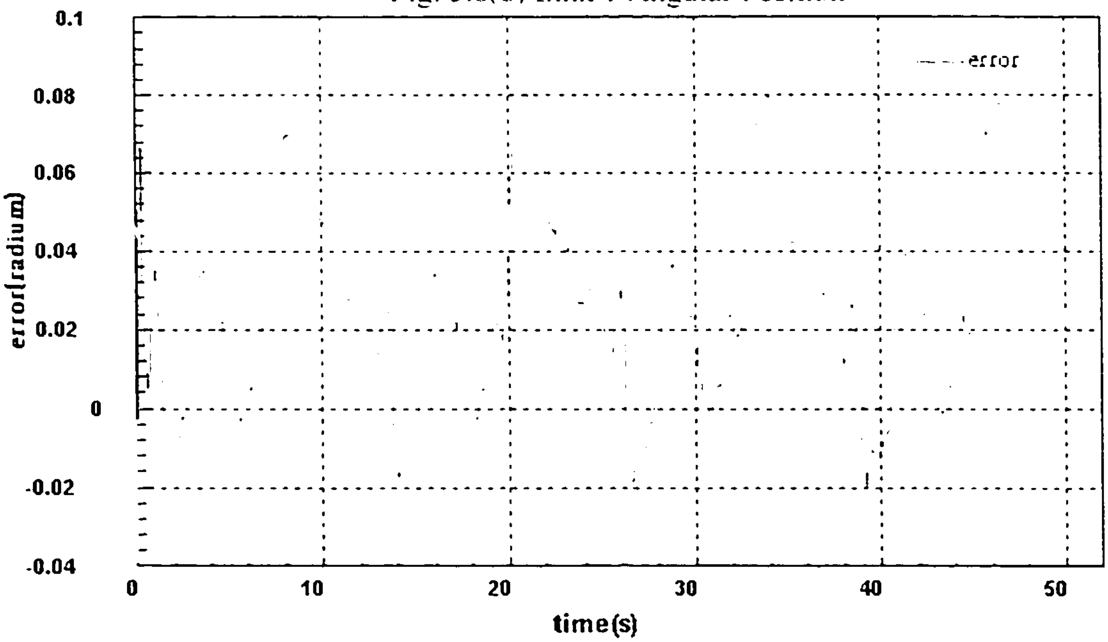

Fig. 5.8(c) Tracking Error

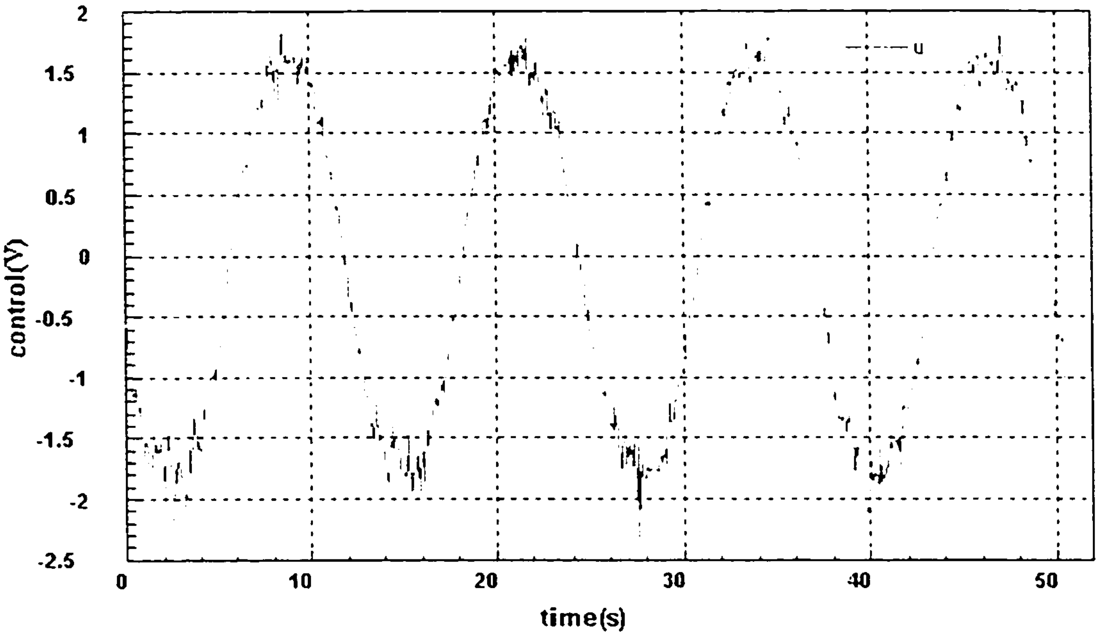

Fig. 5.8(d) Control Signal

Fig. 5.9 shows results when the tracked sinusoidal signal is 180+70 sin0.5t. Fig. 5.9(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 and the dotted line is the desired trajectory to be followed. Fig. 5.9(b) gives the trajectory of q_1. Fig. 5.9(c) is the output tracking error and the Fig. 5.9(d) is the control signal.

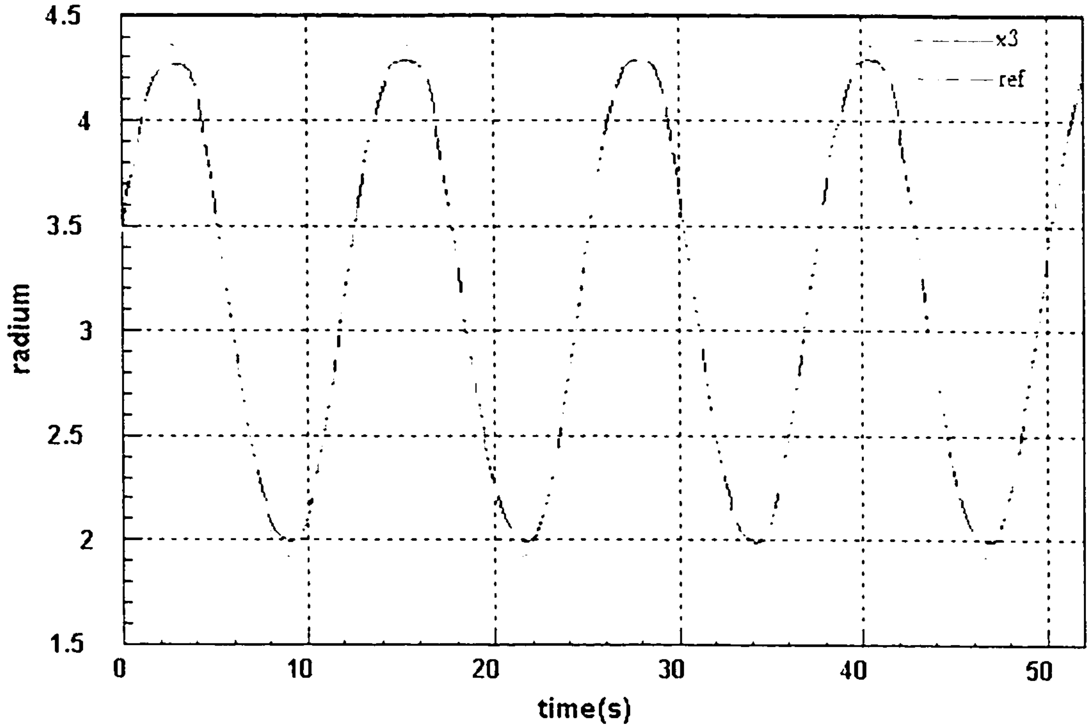

Fig. 5.9(a) Link 2 Angular Position

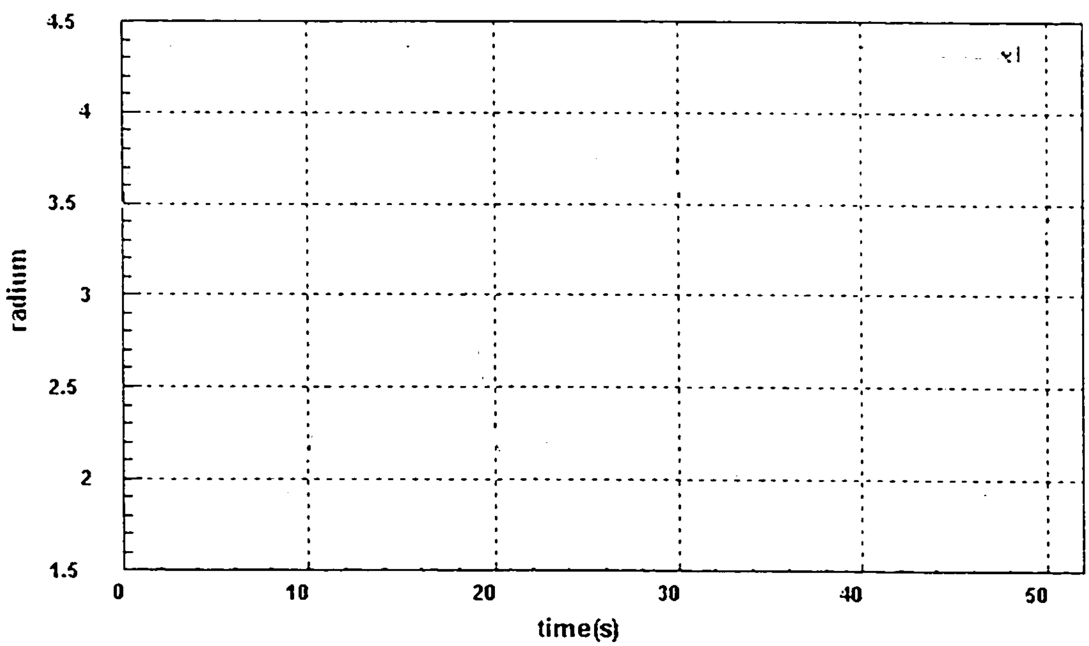

Fig. 5.9(b) Link 1 Angular Position

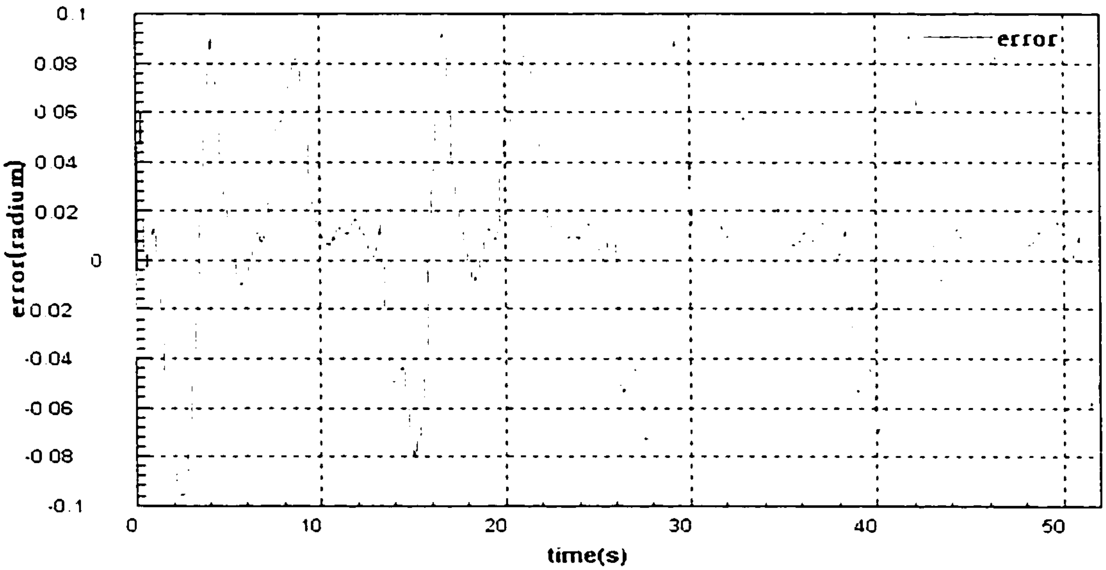

Fig. 5.9(c) Tracking Error

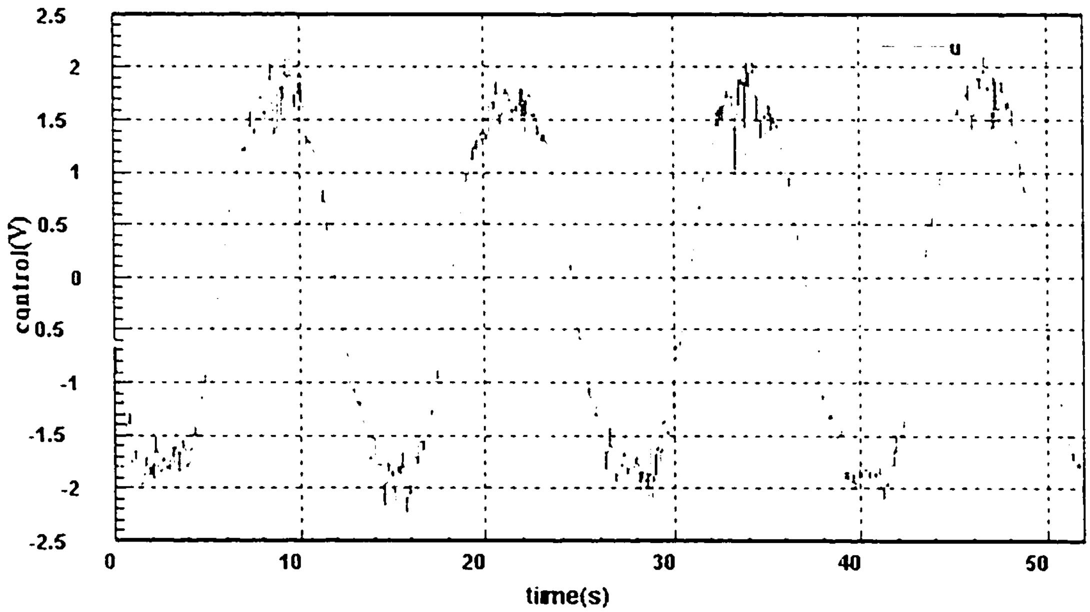

Fig. 5.9(d) Control Signal

Fig. 5.10(a) and (b) are some photos illustrating the real-time trajectory tracking around the mid position.

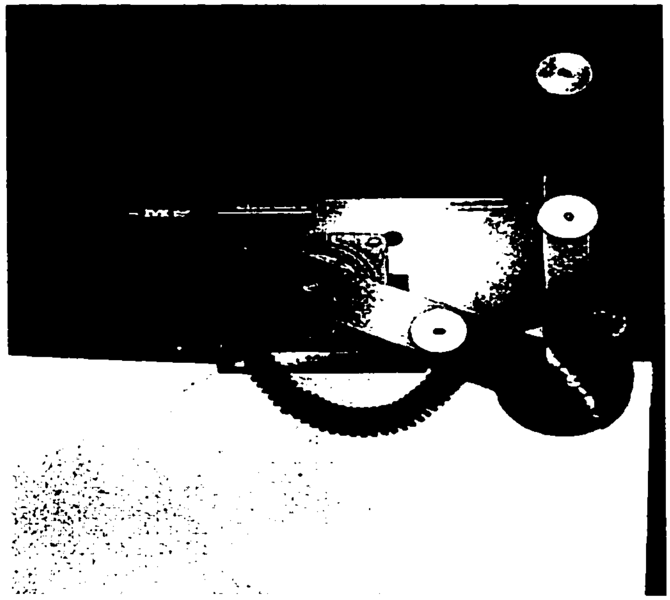

Fig. 5.10(a) trajectory tracking around the Mid Position

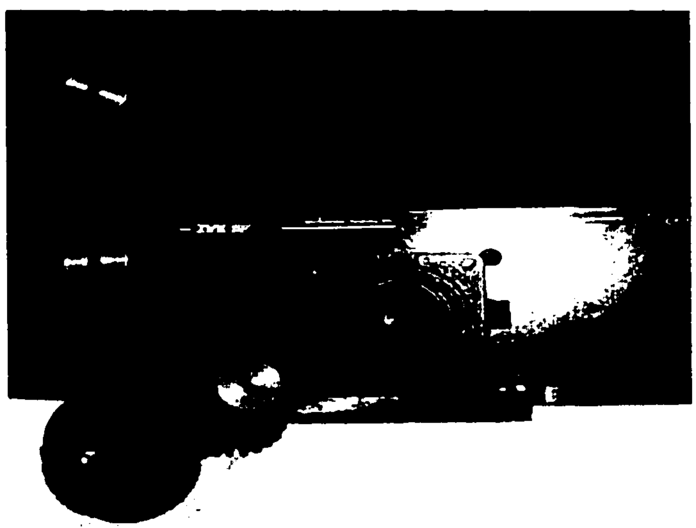

Fig. 5.10(b) Trajectory Tracking around the Mid Position

Fig. 5.7- Fig. 5.9 demonstrate a perfect tracking and Fig. 5.3-Fig. 5.5 also present an acceptable tracking. The closer to the uncontrollable equilibrium the link moves, the harder it is to control them. But from these figures, we can see there is no obvious change of the tracking error when the amplitude of the reference signal changes from 50^0 to 70 and the

amplitude of the reference signal is larger than 40^0. This means that fuzzy control is a sound tool for the control of nonlinear systems.

Finally, in our experiment, a digital computer plays the role of the digital controller. From our experiment, it can be concluded that, for controlling systems with slow dynamics the performance of the digital computer is indistinguishable from the performance give by a continuous controller. However, but for systems that respond at higher frequencies (for example, the design of the estimator) problems in achieving control may be encountered. Therefore the cycle speed, or the sampling period, plays an important role in the design of the digital controller.

Chapter 6

Fuzzy Output Control For Nonlinear Systems

In this chapter, a fuzzy output control scheme for nonlinear systems is presented. This algorithm uses T-S fuzzy rules, whose design for local control laws is based on the linear regulatory theory [22]. For each linear model, an observer driven by the measured output error was constructed for the purpose of asymptotically tracking the linear system and exosystem. This control algorithm is a more commonly used and realistic tool than in the case where all the states are measurable. For a more complete understanding, some well-known preliminaries are presented about the T-S model and linear regulator theory. Based on these preliminaries, a control algorithm is then proposed.

6.1. T-S Fuzzy Control

The Takagi-Sugeno Fuzzy System was discussed in section 4.1. The overall state and output of the fuzzy system is inferred as:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t))(A_i x(t) + B_i u(t))}{\sum_{i=1}^{r} \lambda_i(z(t))}$$

$$y(t) = \frac{\sum_{i=1}^{r} \lambda_i(z(t)) C_i x(t)}{\sum_{i=1}^{r} \lambda_i(t)}$$

6.2. Linear Regulator Theory

Consider a linear system

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) + P\omega(t) \\
\dot{\omega}(t) &= S\omega(t) \\
e(t) &= Cx(t) + Q\omega(t)
\end{aligned}$$

where x is the internal state vector of the plant and is defined in a neighbourhood X of R^n; u is the control input vector; ω is a vector containing external disturbances or references and defined in a neighbourhood W of the origin of R^s. And e is the tracking error if ω is treated as a reference signal.

When the disturbances and/or references $\omega(t)$ are not equal to zero, linear regulator theory can be utilized to ensure the asymptotic output tracking. In the linear regulator theory the control goal is to obtain a stable closed loop system and asymptotic tracking error for every possible exogenous input in a prescribed family of functions of time. This latter requirement is also known as the property of *output regulation.*

To ensure $e(t) \to 0$, the following is assumed:

A1. The pair (A, B) can be stablizabled.

A2. $\sigma(s) \subset clos(c^{+}) = \{\lambda \in c \; \mathrm{Re}[\lambda] \geq 0\}$;

A3. The pair

$$[C \quad Q], \begin{bmatrix} A & P \\ 0 & S \end{bmatrix}$$

is detectable.

A4. Matrices Π and Γ exist and solve the following linear matrix equation

$$\Pi S = A\Pi + B\Gamma + P$$
$$0 = C\Pi + Q \tag{6.1}$$

With assumption (A1), there exists matrix K such that A+BK is stable. With the assumption (A3), there exists matrix G, such that the matrix

$$\begin{pmatrix} A & P \\ 0 & S \end{pmatrix} - \begin{pmatrix} G_0 \\ G_1 \end{pmatrix} (C \quad Q) \tag{6.2}$$

has all eigenvalues in the left-complex plane.

Defining

$$L = \Gamma - K\Pi \tag{6.3}$$

and according to the linear regulator theory, the control law is

$$u = Hz \tag{6.4}$$

$$\dot{z} = Fz + Ge \tag{6.5}$$

and z is defined on a neighborhood Z of the origin of R^v.

where

$$F = \begin{pmatrix} A - G_0 C + BK & P - G_0 Q + BL \\ -G_1 C & S - G_1 Q \end{pmatrix}$$

$$G = \begin{pmatrix} G_0 \\ G_1 \end{pmatrix}$$

$$H = \begin{bmatrix} K & L \end{bmatrix}$$

If we let $\omega = 0$ and substitute the control law (6.4) into the linear system, we obtain:

$$\dot{x} = Ax + BHz$$

$$\dot{z} = Fz + GCx$$

If we let $x_l = \begin{bmatrix} x \\ z \end{bmatrix}$, we obtain:

$$\dot{x}_l = \begin{bmatrix} A & BH \\ GC & F \end{bmatrix} x_l$$
$$= A_l x_l$$

In order to obtain the stable closed loop system from the theory of Linear Regulator [28], we know that A_l is stable, and that there exists a neighborhood of $O \subset X \times Z \times W$ of $(0,0,0)$ such that for each initial condition $(x(0), z(0), \omega(o)) \in O$, $\lim_{t \to \infty} e(t) = 0$.

6.3 Proposed Algorithm

At this stage, a new T-S fuzzy control for the output tracking control of nonlinear systems can be presented. To design the control rules, the same fuzzy sets as antecedents are used in the above plant rules. Consequently, local laws, based on the linear regulator theory for each local linear model, have been designed. The rules for the plant and controller are:

i^{th} **Plant Rule:**

IF $z_1(t)$ is M_{1i}, ... , $z_j(t)$ is M_{ji}, ..., $z_q(t)$ is M_{qi},

$$\text{Then} \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) + P_i \omega(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = C_i x(t) + Q_i \omega(t) \end{cases} \tag{6.6}$$

i^{th} **Controller Rule:**

IF $z_1(t)$ is M_{1i}, ... , $z_j(t)$ is M_{ji}, ... , $z_q(t)$ is M_{qi}

$$\text{Then} \begin{cases} \dot{z} = F_i z + G_i e \\ u = H_i z \end{cases} \tag{6.7}$$

where H_i, F_i, G_i is the control matrix for the each $(A_i, B_i, C_i, P_i, S, Q_i)$, $i = 1, \ldots, r$

In this case, the final state and tracking error of the fuzzy system is inferred as follows:

$$\dot{x}(t)=\frac{\sum_{i=1}^{r}\lambda_i(z(t))(A_i x(t)+B_i u(t)+P_i\omega(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}$$

$$e(t)=\frac{\sum_{i=1}^{r}\lambda_i(z(t))(C_i x(t)+Q_i\omega(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}$$

Hence the overall fuzzy controller is given by

$$u(t)=\frac{\sum_{i=1}^{r}\lambda_i(z(t))[H_i Z(t)]}{\sum_{i=1}^{r}\lambda_i(z(t))} \tag{6.8}$$

The design purpose in this study is to specify the fuzzy control in (6.8) to achieve a global stable and tracking control. The following result was obtained:

Theorem: If the local linear system in (6.6) satisfies Assumptions A1-A4, and the local output regulator (6.7) can asymptotically stabilize the local linear system in (6.6) and ensure the asymptotic trajectory tracking for every possible initial state and every possible exogeneous input, then, the fuzzy control (6.8) can globally stabilize the Takagi-Sugeno system (6.6) and ensures the asymptotic trajectory tracking.

Proof: Substituting (6.8) into (6.6) and combining (6.7), we obtain:

$$\dot{x}=\sum_{i=1}^{r}A_i N_i x+\sum_{i=1}^{r}B_i N_i\left(\sum_{j=1}^{r}H_j N_j z\right)+\sum_{i=1}^{r}P_i N_i\omega$$

$$\dot{z}=\sum_{i=1}^{r}F_i N_i z+\sum_{i=1}^{r}G_i N_i\left(\sum_{j=1}^{r}C_j N_j\right)x+\sum_{i=1}^{r}G_i N_i\left(\sum_{j=1}^{r}Q_j N_j\right)\omega$$

where

$$N_i=\frac{\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\qquad N_j=\frac{\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}$$

If we let $\omega = 0$, we obtain:

$$\dot{x} = \sum_{i=1}^{r} A_i N_i x + \sum_{i=1}^{r} B_i N_i \left(\sum_{j=1}^{r} H_j N_j z \right)$$

$$\dot{z} = \sum_{j=1}^{r} F_j N_j z + \sum_{j=1}^{r} G_j N_j \left(\sum_{i=1}^{r} C_i N_i \right) x$$

If we let $x_l = \begin{bmatrix} x \\ z \end{bmatrix}$, we can obtain a new state equation:

$$\dot{x}_l = \begin{bmatrix} \sum_{i=1}^{r} A_i N_i & \sum_{i=1}^{r} B_i N_i \left(\sum_{j=1}^{r} H_j N_j \right) \\ \sum_{j=1}^{r} G_j N_j \left(\sum_{i=1}^{r} C_i N_i \right) & \sum_{j=1}^{r} F_j N_j \end{bmatrix} x_l$$

In order to prove that the (x, z) of (0, 0) of the Takagi-Sugeno Fuzzy scheme is globally asymptotically stable, the direct method of Lyapunov was used. A (quadratic) Lyapunov function is chosen

$$V(x_l) = x_l^T P_l x_l$$

where P_l is a "positive definite matrix" (denoted by $P_l>0$) that is symmetric (i.e., $P_l=P_l^T$). If P_l is positive definite, then for all $x_l \neq 0$, $x_l^T P_l x_l > 0$. Hence, we have $V(x_l)>0$ and $V(x_l)=0$ only if $x_l=0$. Also $|x_l| \to \infty$, then $V(x_l) \to \infty$.

To show that the equilibrium $x_l = 0$ of the closed loop system is globally asymptotically stable, we need to show that $\dot{V}(x_l) < 0$ for all x_l. Notice that

$$\dot{V}(x_l) = x_l^T P_l \dot{x}_l + \dot{x}_l^T P_l x_l$$

so that since

$$N_i = \frac{\lambda_i(z(t))}{\sum_{i=1}^{r} \lambda_i(z(t))}, \quad N_j = \frac{z(t)}{\sum_{j=1}^{r} \lambda_j(z(t))}$$

the derivative of the $V(x_l)$ can be written in the following forms:

$$\dot{V}(x_l) = x_l^T P_l
\begin{bmatrix}
\dfrac{\sum_{i=1}^{r} A_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))} & \dfrac{\sum_{i=1}^{r} B_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\left(\dfrac{\sum_{j=1}^{r} H_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\right) \\
\dfrac{\sum_{j=1}^{r} G_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\left(\dfrac{\sum_{i=1}^{r} C_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\right) & \dfrac{\sum_{j=1}^{r} F_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}
\end{bmatrix} x_l + x_l^T$$

$$\begin{bmatrix}
\dfrac{\sum_{i=1}^{r} A_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))} & \dfrac{\sum_{i=1}^{r} B_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\left(\dfrac{\sum_{j=1}^{r} H_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\right) \\
\dfrac{\sum_{j=1}^{r} G_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\left(\dfrac{\sum_{i=1}^{r} C_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\right) & \dfrac{\sum_{j=1}^{r} F_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}
\end{bmatrix}^T P_l x_l$$

$$\dot{V}(x_l) = x_l^T P_l
\begin{bmatrix}
\dfrac{\sum_{i=1}^{r} A_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\dfrac{\sum_{j=1}^{r}\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))} & \dfrac{\sum_{i=1}^{r} B_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\left(\dfrac{\sum_{j=1}^{r} H_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\right) \\
\dfrac{\sum_{j=1}^{r} G_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\left(\dfrac{\sum_{i=1}^{r} C_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\right) & \dfrac{\sum_{j=1}^{r} F_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\dfrac{\sum_{i=1}^{r}\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}
\end{bmatrix} x_l + x_l^T$$

$$\begin{bmatrix}
\dfrac{\sum_{i=1}^{r} A_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\dfrac{\sum_{j=1}^{r}\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))} & \dfrac{\sum_{i=1}^{r} B_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\left(\dfrac{\sum_{j=1}^{r} H_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\right) \\
\dfrac{\sum_{j=1}^{r} G_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\left(\dfrac{\sum_{i=1}^{r} C_i\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}\right) & \dfrac{\sum_{j=1}^{r} F_j\lambda_j(z(t))}{\sum_{j=1}^{r}\lambda_j(z(t))}\dfrac{\sum_{i=1}^{r}\lambda_i(z(t))}{\sum_{i=1}^{r}\lambda_i(z(t))}
\end{bmatrix}^T P_l x_l$$

By denoting the sum over all possible combinations of $\left(\sum_{i=1}^{r}\lambda_i(z(t))\right)\left(\sum_{j=1}^{r}\lambda_j(z(t))\right)$ as $\sum_{i,j}$,

we obtain:

$$
\begin{aligned}
\dot{V}(x_l) &= x_l^T P_l
\begin{bmatrix}
\dfrac{\sum_{i,j} A_i\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))} & \dfrac{\sum_{i,j} B_i H_j\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))} \\
\dfrac{\sum_{i,j} G_j C_i\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))} & \dfrac{\sum_{i,j} F_j H_i\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))}
\end{bmatrix} x_l + x_l^T \\
&\begin{bmatrix}
\dfrac{\sum_{i,j} A_i\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))} & \dfrac{\sum_{i,j} B_i H_j\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))} \\
\dfrac{\sum_{i,j} G_j C_i\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))} & \dfrac{\sum_{i,j} F_j H_i\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j}\lambda_i(z(t))\lambda_j(z(t))}
\end{bmatrix}^T P_l x_l
\end{aligned}
$$

Since:

$$
0 \le \frac{\lambda_i(z(t))\lambda_j(z(t))}{\sum_{i,j=1}^{r}\lambda_i(z(t))\lambda_j(z(t))} \le 1
$$

we have

$$
\begin{aligned}
\dot{V}(x_l) &\le x_l^T P_l \begin{bmatrix} \sum_{i,j} A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_{i,j} F_j \end{bmatrix} x_l + x_l^T \begin{bmatrix} \sum_{i,j} A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_{i,j} F_j \end{bmatrix}^T P_l x_l \\
&= x_l^T \left\{ \begin{aligned} & P_l \begin{bmatrix} \sum_{i,j} A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_{i,j} F_j \end{bmatrix} + \\ & \begin{bmatrix} \sum_{i,j} A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_{i,j} F_j \end{bmatrix}^T P_l \end{aligned} \right\} x_l
\end{aligned}
$$

From the Linear Regulator Theory, we know that $\begin{bmatrix} A_i & B_i H_j \\ G_j C_i & F_j \end{bmatrix}$ is stable, so $\begin{bmatrix} \sum_i A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_j F_j \end{bmatrix}$ is stable. From the theory of linear matrix inequality, we know that we can find a positive definite matrix P_l to let

$P_i\begin{bmatrix}\sum_i A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_i F_i\end{bmatrix}+\begin{bmatrix}\sum_i A_i & \sum_{i,j} B_i H_j \\ \sum_{i,j} G_j C_i & \sum_i F_i\end{bmatrix}^T P_i < 0$ Hence, we have proved that $\dot{V}(x_i) < 0$ and the equilibrium (x.z)=(0.0) is globally asymptotically stable in the first approximation.

When $\omega(t) \neq 0$, it can be shown that $\underset{t\to\infty}{Lim}\, e(t) = 0$.

$$
\begin{aligned}
e(t) &= \frac{\sum_{i=1}^{r} \lambda_i(z(t)) C_i x(t)}{\sum_{i=1}^{r} \lambda_i(z(t))} + \frac{\sum_{i=1}^{r} \lambda_i(z(t)) Q_i \omega(t)}{\sum_{i=1}^{r} \lambda_i(z(t))} \\
&= \frac{\sum_{i=1}^{r} \lambda_i(z(t))(C_i x(t) + Q_i \omega(t))}{\sum_{i=1}^{r} \lambda_i(z(t))}
\end{aligned}
$$

Because $0 \leq \dfrac{\lambda_i(z(t))}{\sum_{i=1}^{r} \lambda_i(z(t))} \leq 1$, $0 \leq e(t) \leq \sum_{i=1}^{r}(C_i x(t) + Q_i \omega(t)) = \sum_{i=1}^{r}(e_i(t))$

And because $\underset{t\to\infty}{Lim}\, e_i(t) = 0$, $\underset{t\to\infty}{Lim}\, e(t) = 0$. The proof is finished.

Chapter 7

Simulation Results

7.1. Linearized Model

As in section 5.1, the linearized system for the Pendubot is:

$$\dot{x} = Ax(t) + Bu(t)$$

where $x := \delta x$, $A := F_x(t) = \frac{\partial F(x(t),u(t),t)}{\partial x}\Big|_{x^0,u^0}$ $B := \frac{\partial F(x(t),u(t),t)}{\partial u}\Big|_{x^0,u^0} = F_u(t)$

7.2. The application of the control law

In this section, the simulation of the proposed algorithm to the Pendubot in chapter 6 is discussed. The objective is to force the angular position of link 2, q_2, to follow a sinusoidal signal of 50^0 amplitude. In order to track this signal, at every trajectory point the equality $q_1 + q_2 = 90^0$ must be filled.

Fuzzy plant: To model the Pendubot, the five fuzzy sets are proposed in Fig. 7.1. The notation is BL (Big Larger PI), ML (Medium Larger PI), E (Equal to PI), MS (Medium Smaller PI), BS (Big Smaller PI). To obtain the linear models for the consequents, the nonlinear model of Pendubot is linearized around the following equilibrium points

$$x_i^0 = \left[q_{1i}^0, 0, q_{2i}^0, 0\right],\ i = 1,\dots,5$$

where $q_{21}^0 = 130^0$, $q_{22}^0 = 155^0$, $q_{23}^0 = 180^0$, $q_{24}^0 = 205^0$, $q_{25}^0 = 230^0$ and $q_{1i}^0 = 90^0 - q_{2i}^0$, $i = 1,\dots,5$

For each equilibrium point a linear system (A_{ei}, B_{ei}), $i = 1,\dots,5$ is obtained. The values of (A_{ei}, B_{ei}) are in appendix G. Since the output y is q_2, C=[0 0 1 0] is valid for all local systems. The external perturbation is not considered for this system, hence the matrix P=0.

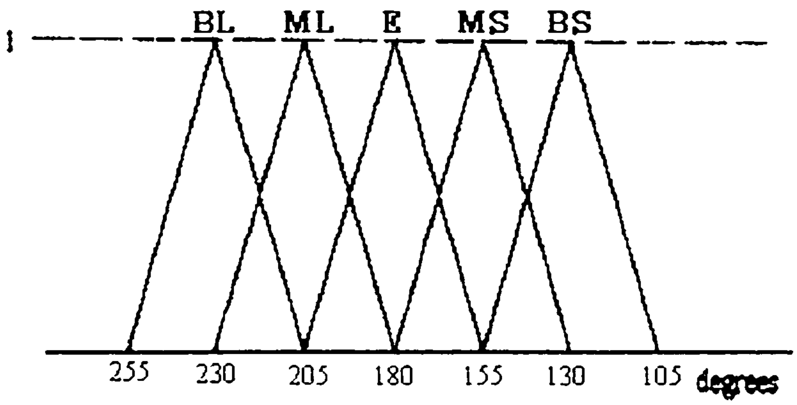

Fig.7.1 FUZZY SETS

The reference signal and the exosystem are the same as those in the section 5.2.2 control. The reference has to be generated.

$y_r = u_{opi} + k\sin\omega t; \qquad i = 1,\dots,5$

Based on the above y_r, the exosystem for each linear region is thus expressed as:

$$\dot{\omega}(t) = S\omega(t);$$
$$y_r(t) = Q_i\omega(t)$$

For each $(A_{ei}, B_{ei}, C_{ei}, P_{ei}, S, Q_{ei})$ the matrices Γ_i and Π_i in the fuzzy controller are determined from (7.1). We use the "Maple" to calculate them. The L_i is calculated using (7.2). each K_i and G_i is selected according to the Pole Placement Method. In this thesis. each K_i is selected to have the eigenvalues of $(A_{ei}+B_{ei}K_i)$ located at

$$[-5.2\ -6.8\ -5.4\ -6.6]$$

For the linear system (A_{e1}, B_{e1}) and (A_{e5}, B_{e5}). each G_i is selected to have the eigenvalues of (7.3) located at

$$[-17.5\ -35.5\ -18\ -38.5\ -0.3\ -2\ -1.4]$$

For the linear system (A_{e2}, B_{e2}) and (A_{e4}, B_{e4}), each G_i is selected to have the eigenvalues of (7.3) located at

$$[-17.5\ -39.5\ -18\ -38.5\ -0.1\ -2\ -1.5]$$

For the linear system (A_{e3}, B_{e3}), Gi is selected to have the eigenvalues of (7.3) located at

$$[-17\ \ -39.6\ \ -18.6\ -38.4\ -1.8\ \ -1.5]$$

Each F_i and H_i is calculated using (7.4) and (7.5). We use the "Matlab" to calculate them(Appendix H). The particular values of F, G and H for each local controller are given in appendix G. The local control signals are obtained from (7.7). Finally the control signal applied to the system is given by (7.8).

Fuzzy Rules: There are five fuzzy rules for the plant and for the controller.

1st Plant Rule

If q_2 is big smaller PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_{e1}x(t) + B_{e1}u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_{e1}\omega(t) \end{cases}$$

1st Controller Rule

If $q_2(t)$ is big smaller PI.

$$\text{Then}\begin{cases} \dot{z} = F_1 z + G_1 e \\ u = H_1 z \end{cases}$$

2nd Plant Rule

If q_2 is medium smaller PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_{e2}x(t) + B_{e2}u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_{e2}\omega(t) \end{cases}$$

2^{nd} Controller Rule

If $q_2(t)$ is medium smaller PI.

$$\text{Then}\begin{cases} \dot{z} = F_2 z + G_2 e \\ u = H_2 z \end{cases}$$

3^{rd} Plant Rule

If q_2 is equal to PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_{e3}x(t) + B_{e3}u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_{e3}\omega(t) \end{cases}$$

3^{rd} Controller Rule

If $q_2(t)$ is equal to PI.

$$\text{Then}\begin{cases} \dot{z} = F_3 z + G_3 e \\ u = H_3 z \end{cases}$$

4^{th} Plant Rule

If q_2 is medium larger PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_{m4}x(t) + B_{m4}u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_{m4}\omega(t) \end{cases}$$

4^{th} Controller Rule

If $q_2(t)$ is medium larger PI.

$$\text{Then}\begin{cases} \dot{z} = F_4 z + G_4 e \\ u = H_4 z \end{cases}$$

5^{th} Plant Rule

If q_2 is big smaller PI

$$\text{Then}\begin{cases} \dot{x}(t) = A_{e5}x(t) + B_{e5}u(t) \\ \dot{\omega}(t) = S\omega(t) \\ e(t) = Cx(t) + Q_{e5}\omega(t) \end{cases}$$

5^{th} Controller Rule

If $q_2(t)$ is big smaller PI.

$$\text{Then}\begin{cases} \dot{z} = F_5 z + G_5 e \\ u = Hz \end{cases}$$

7.3.Simulation Results

Fig. 7.2 illustrated the results when the tracked sinusoidal signal is 180+50sin0.5t. Fig. 7.2(a) represents the tracking performance of q_2 where the line is the actual trajectory of q_2 and the dotted line is the desired trajectory to be followed. Fig. 7.2(b) gives the trajectory of q_1. Fig. 7.2(c) is the output tracking error and the Fig. 7.2(d) the control signal. We simulate it in C code (Appendix I). In this case, the closed loop system presents good tracking without control signal saturations.

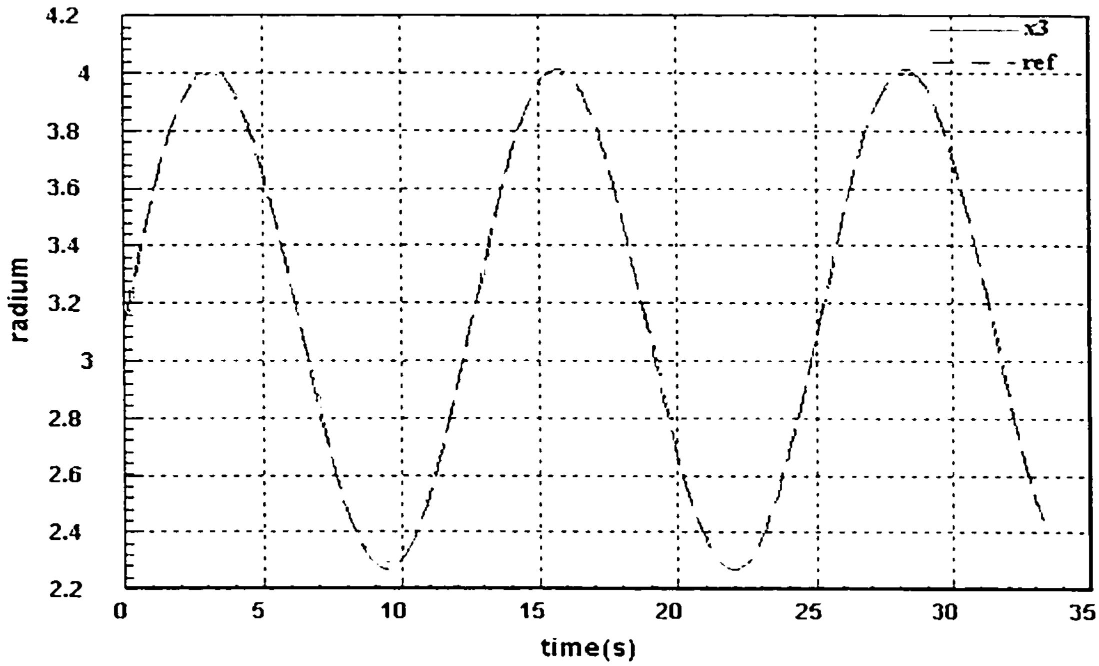

Fig. 7.2(a) Link 1 Angular Position

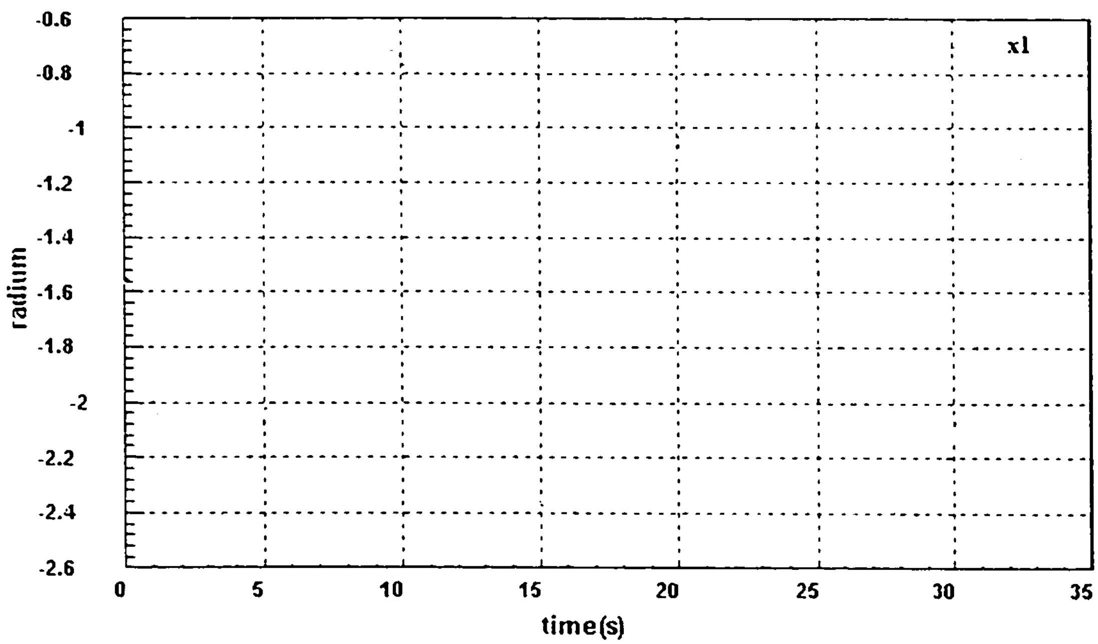

Fig. 7.2(b) Link 2 Angular Position

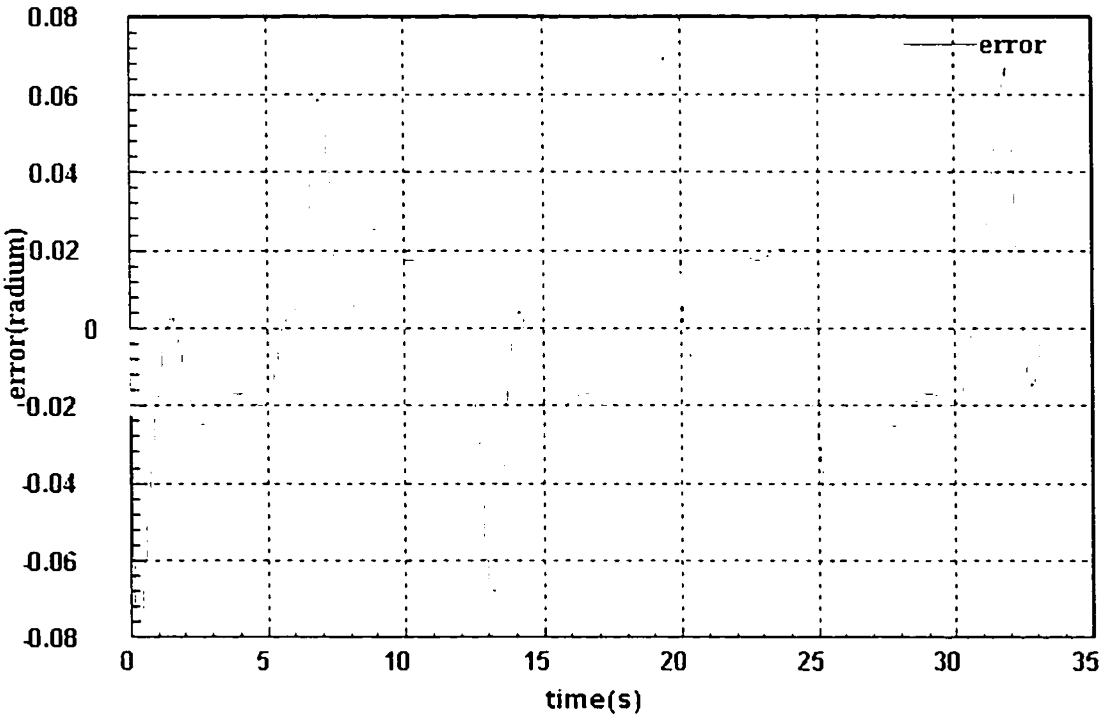

Fig. 7.2(c) Tracking Error

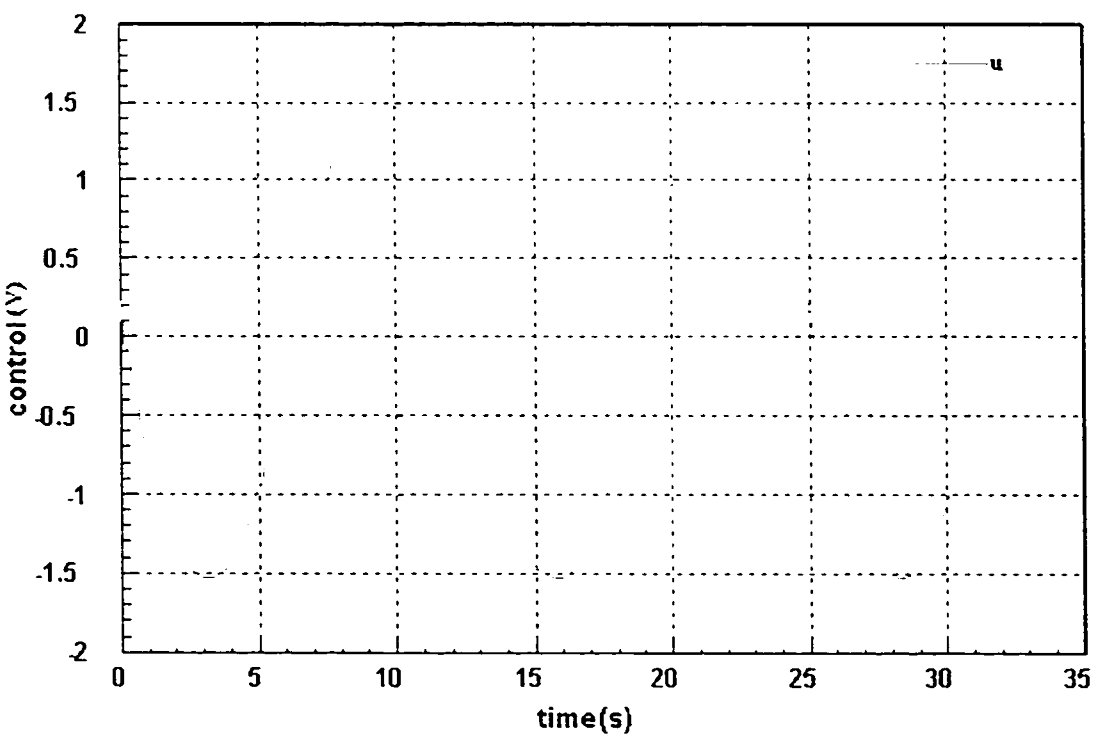

Fig. 7.2(d) Control Signal

The advantage of the proposed control design is that only a simple fuzzy controller is used as an alternative to a more complex design. And the simulation result confirms the accurate output tracking capability and robust performance of the proposed method.

Chapter 8

Summary and Recommendations for Future Work

8.1. Discussion

The study of under-actuated mechanisms is a topic of currently research topic in the control community. These systems are characterized by the fact that they have fewer independent control actuators than degrees of freedom to be controlled. Different control schemes, have been proposed, derived from adaptive control theory [29], classical control techniques [1] [6], or intelligent control methodologies [30] [31], have been proposed. From the recent research papers, it is found that the fuzzy control method attracts many researchers because of its strong approximation.

In brief, the fuzzy control method used in the Pendubot can be divided into several procedures. First, the nonlinear system is converted into a set of linear local system model at each equilibrium point using the Taylor series. The second procedure is to design a linear controller based on the theory of linear regulator or linear optimal theory for each local linear model. Thirdly, combine all linear controllers with the Takagi-Sugeno fuzzy schemes to result in an overall fuzzy controller. In this thesis, a triangle membership function was used, and it has been proven that this membership function works very well.

Other research to the under-actuated mechanism includes the following aspects.

1. *Swing up and Balance*. This is a set-point regulation, and two classical control algorithms (partial feedback linearization and state feeback) were used to implement them. The experimental result is very good.

2. *Switching Control*. So-called logic based switching control is widely proposed and here the logic condition may be the error of angle or the distance to the goal position [2]. The error of angle was used in this thesis.

3. *Trajectory Tracking*. This is a new study aimed specifically at the under-actuated mechanism. The fuzzy algorithm used for optimal trajectory tracking control can be thought of as the main contribution of this thesis.

8.2. Conclusions

In our study, a fuzzy scheme for optimal trajectory tracking control is proposed and implemented in the actual under-actuated mechanism-the Pendubot. Compared with the previous method [2], it is found that the tracking error is very small and the stability is excellent. The Pendubot can be stablized in a large offset from the top and the mid positions. This method is simpler than the classical method [2], and the tuning is easy for the experiment. This proposal method represents a new class of the fuzzy controls. The real-time implementation also shows the success of the application and feasibility of the fuzzy method for the nonlinear system.

Another fuzzy control algorithm that uses the output feedback is also proposed as an alternative approach for trajectory tracking. This control algorithm is a more realistic for common situation. It is not necessary to assume the state is completely measured. The simulation was given to verify this scheme. In this simulation, the sampling time is 0.001 s. The smaller the sample time is, the more accurate the mathematical model is.

The stability analysis of these two fuzzy algorithms for trajectory tracking of nonlinear systems is also important. The possibility and validity of these two fuzzy schemes was verified. The stability proof suggests that we can apply linear control theory and fuzzy control method to research the nonlinear systems.

8.3 Recommendations for future work

8.3.1. Generalization

The fuzzy control scheme developed in this thesis has been successfully applied to the Pendubot, an example of the under-actuated mechanisms. Therefore, the future work should focus on the generation of this fuzzy control method for general under-actuated systems. Furthermore, in order to generalize this control method, the digital filter (used for the approximation of the joint velocities) should be designed in the computer program to search for a best control effort.

8.3.2. Trajectory Tracking

For the nonliner system, trajectory tracking is more difficult than the set-point regulation. The research results from [2] and [21] are limited. In this thesis, this is the primary contribution to studies of this kind. For the output feedback fuzzy scheme, a good design procedure for the estimator to implement it in the real time needs to be pursued. The Off-Equilibrium Linearisation Methods [32] in Gain-Scheduling can be used to model nonlinear system. It can be combined with the linear regulator theory to realize the output tracking of nonlinear system.

References

[1] Lsabelle Fand, Regolio Lozano, and Mark W.Spong, "Passivity based on control o f the Pendubot", *Proceedings of the American Control Conference San Diego*, California, June 1999.

[2] Daniel Jerome Block, "Mechanical Design and Control of the Pendubot", M.S.Thesis, Department of General Engineering, University of Illionis at Urbana-champain, 1996

[3] Mark W.Spong "The Swing Up Control Problem for The Acrobot", *IEEE Transactions on Control Systems Magazine*, pp. 49-55, February 1995.

[4] Christian Melin, Borislav Vidolov, "A fuzzy PD-Like Scheme for Two underactuated Planar Mechanisms", *Proceedings of IEEE International Fuzzy Systems Conference*, 2000.

[5] Marcelo C.M., Teixeira, Stanislaw and H.Zak, "Stabilizing Controller Design for Uncertain Nonlinear Systems Using Fuzzy Models", *IEEE Transactions on Fuzzy System*, vol. 7, no. 2, April 1999

[6] Rafael Fierro, Frank L.Lewis, and Andy Lowe, "Hybrid Control for a class of underactuated Mechanical Systems", *IEEE Transactions on System, Man, and Cybernetics--Part A: System and Humans*, vol.29, no.6, November 1999.

[7] Kazunobu Yoshiba, "Swing-Up Control of an Inverted Pendulum by Energy-based Methods", *Proceedings of the American Control Conference, San Diego*, California June, 1999.

[8] Matthew D.Berkemeier and Ronald S. Fearing, "Tracking Fast Inverted Trajectories of the Underactuated Acrobot", *IEEE Transactions on Robotics and Automation*, vol. 15,

no. 4, August 1999.

[9] Jianqiang Yi, Naoyoshi Yubazaki and Kaoru Hirota, "Systematically Constructing Stabilization Fuzzy Controllers for Single and Double Pendulum Systems", *IEEE Conference on Fuzzy System*, 2000.

[10] Edgar Sanchez, Luis A.Nuno, Ya-Chan Hsu, and Guanrong Chen, " Fuzzy PD Scheme for Underactuated Robot Swing-up Control", *Proceedings of IEEE International Fuzzy System*, 1998.

[11] N.S. Dedrossian, "Nonlinear Control of an Under-actuated Two link Manipulator", *Proceedings of the 32nd Conference on Decision and Control*, pp. 2234-2238, 1993.

[12] H.Arai and S. Tachi, "Position control of a manipulator with passive joints using dynamic coupling", *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 528-534 1991.

[13] H.Arai, "Controllability of a 3-DOF manipulator with a passive joint under a nonholonomic constraint", *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 3707-3713,1996.

[14] Ju-Jang Lee, "Robust Adaptive Control of an Underactuated Free-flying Space Robot Under a Non-holonomic Structure in Joint Space", *International Journal of Systems Science*, vol. 27, no. 11, pp. 1113-1221,1996.

[15] Chun-Yi Su and Yury Stepaneko, "Sliding Model Control of Nonholonomic Mechanical Systems: Under-actuated Manipulator Case", *Nonlinear Control System Design Symposiums (NOLCOS), IFAC Preprints*, Tahac city, pp. 609-613, 1995.

[16] G.Oriolo and Y.Nakamura. " Control of Mechanical Systems with Second-order Non - holonomic Constraints: Under-actuated Manipulators". *Proceeding of the 30^{th} Conference on Decision and Control*. pp. 2398-2403. 1991.

[17] T.Takagi and M.Sugeno. "Fuzzy idenfication of systems and its applications modeling and control". *IEEE transactions on Systems, Man and Cybernitics*. vol. 15. no. 1. pp.116-132.1985.

[18] K.Tanaka and M.Sugeno. "Stability analysis and design of fuzzy control system". *Fuzzy Sets and Systems*. vol. 45. no. 2. pp.133-156. 1992.

[19] K.Tanaka. T.Taniguchi. and H.O.Wang. "Fuzzy control based on quadratic Performance function: a linear matrix inequality approach". *Proceedings of IEEE Conference on Decision and Control*. Tampa. Florida. USA. December 1998.

[20] T.Taniguchi. K.Tanaka. K.Yamafuji and H.O.Wang. "Nonlinear model following control via Takagi-Sugeno fuzzy model". *Proceedings of the American Control Conference*. San Diego. California. June 1999.

[21] O. Begovich. E. N. Sanchez and M. Maldonado. "T-S Scheme forTrajectory Tracking of An Underactuated Robot". *Proceedings of IEEE International Conference on Fuzzy Systems*. vol. 2. pp. 798-803. 2000.

[22] B. A. Francis. "The linear multivariable regulator problem". *SIAM J.Control Optim.* vol. 14. pp. 486-505. 1977.

[23] A.Isidori. "Nonlinear Control Systems". *3^{rd} ed. Springer-Verlag*. 1995.

[24] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. pp. 145-148. 1989.

[25] Spong. M.W.. "The Control of Underactuated Mechanical Systems". *Plenary Lecture at the first International conference on Mechatronics*. Mexico City. January 26-29. 1994

[26] Spong. M.W., "Swing Up Control of the Acrobot using Partial Feedback Linearization" *SY-ROCO'94*, Capri, Italy, pp. 833-838, September, 1994.

[27] Shing-Jen Wu and Chin-Teng Lin., "Optimal Fuzzy Controller Design: Local Concept Approach," *IEEE Transactions on Fuzzy System*, vol. 8, no. 2 April, 2000.

[28] A.Isidori, C.I.Bynes and F.Flockerzi "Topics in Control Theory", *Berkhauser* 1994.

[29] Chun-Yi Su and Yury Stepanenko "Adaptive Variable Structure Set-Point Control of Underactuated Robots" *IEEE Transactions on Automatic Control*, vol. 44, no. 11, 1999

[30] Marco A.T.de Sousa and Marconi K.Madrid, "Optimization of Takagi-Sugeno Fuzzy Controllers Using a Genetic Algorithm", *Proceedings of IEEE International Fuzzy Systems Conference*, 2000.

[31] M.A.Lee and M.H.Smith, "Automatic Design and Tuning of a Fuzzy System for controlling the Acrobot using Genetic Algorithms, DSFS, and Meta-Rule Techniques", Proc. MAFIPS'94, San Antonio, Texas, December 1994.

[32] P.Korba, "A Gain-Scheduling Approach to Model-Based Fuzzy Control", *VDI Verlag*, Dusseldorf, Germany, 2000.

[33] Chung-Shi Tseng, Bor-Sen Chen and Huey-Jian Uang "Fuzzy Tracking Control Design or Nonlinear Dynamic Systems via T-S Fuzzy Model", *IEEE Transactions on Fuzzy System*, pp. 381-392, vol. 9, no. 3, June 2001.

Publications

Referenced Conference and Journal Paper:

[1] Zhen Cai and Chun-Yi Su, "Optimal Fuzzy Control of Nonlinear Systems with Application to An Underactuated Robot" accepted for *the 15th IFAC World Cogress*, 2002

[2] Zhen Cai and Chun-Yi Su, "Output Tracking Control of Takagi-Sugeno Fuzzy Systems with application to an Underactuated Robot " accepted for *the IEEE International Conference on Fuzzy Systems*, 2002.

[3] Zhen Cai and Chun-Yi Su, "An optimal real-time trajectory tracking control design for Pendubot via Takagi-Sugeno fuzzy model" submitted to *ASME Journal of Dynamic Systems, Measurement, and Control.*

Appendix A

Linearized Equations

$$f_d(x,u) = f_d(x_r,u_r) + \frac{\partial f}{\partial x}\bigg|_{x_r,u_r}(x - x_r) + \frac{\partial f}{\partial u}\bigg|_{x_r,u_r}(u - u_r)$$

$$u_r = \theta_1 g\cos(x_{r1})$$

$$x_{r1} + x_{r3} = \frac{\pi}{2}$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dfrac{\partial f_2}{\partial x_3} & \dfrac{\partial f_2}{\partial x_4} \\ 0 & 0 & 0 & 1 \\ \dfrac{\partial f_4}{\partial x_1} & \dfrac{\partial f_4}{\partial x_2} & \dfrac{\partial f_4}{\partial x_3} & \dfrac{\partial f_4}{\partial x_4} \end{bmatrix}$$

$$\frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \dfrac{\partial f_2}{\partial u} \\ 0 \\ \dfrac{\partial f_4}{\partial u} \end{bmatrix}$$

$$\frac{\partial f_2}{\partial x_1} = \frac{\theta_2 g(-\theta_4\sin(x_{r1}) - \theta_3\sin(x_{r1} + x_{r3}))}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \frac{(\theta_2 + \theta_3\cos(x_{r3}))\theta_3 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2}$$

$$\frac{\partial f_2}{\partial x_2} = \frac{-2(\theta_2 + \theta_3\cos(x_{r3}))\theta_3\sin(x_{r3})x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} - \frac{2\theta_2\theta_3\sin(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \frac{0.00545\theta_2}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2}$$

$$
\begin{aligned}
\frac{\partial f_2}{\partial x_3} = {} & \frac{-\theta_2 T\theta_3^2 \sin 2x_{r3}}{\left(-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2\right)^2} - \frac{2\theta_2\theta_3^3 \sin(x_{r3})^2 x_{r4}\cos(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2\right)^2} + \frac{\theta_2\theta_3 \cos(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} - \\
& \frac{\theta_3^2 \sin(x_{r3})^2 x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \frac{2(\theta_2 + \theta_3\cos(x_{r3}))\theta_3^3 \sin(x_{r3})^2 x_{r2}\cos(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2\right)^2} + \\
& \frac{(\theta_2 + \theta_3\cos(x_{r3}))\theta_3\cos(x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \frac{\theta_2(-\theta_3\sin(x_{r3})x_{r4} - \theta_3\sin(x_{r3})x_{r2})x_{r4}\theta_3^2\sin(2x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2\right)^2} + \\
& \frac{\theta_2(-\theta_3\cos(x_{r3})x_{r4} - \theta_3\cos(x_{r3})x_{r2})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \\
& \frac{2\theta_2(\theta_4 g\cos(x_{r1}) + \theta_5 g\cos(x_{r1} + x_{r3}))\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2\right)^2} - \\
& \frac{\theta_2\theta_5 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \frac{\theta_3\sin(x_{r3})\theta_5 g\cos(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} - \\
& \frac{2(\theta_2 + \theta_3\cos(x_{r3}))\theta_5 g\cos(x_{r1} + x_{r3})\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2\right)^2} - \\
& \frac{(\theta_2 + \theta_3\cos(x_{r3}))\theta_5 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \\
& \frac{0.01109\theta_2 x_{r2}\theta_3^2\cos(x_{r3})\sin(x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} + \frac{0.00047\theta_3\sin(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2} - \\
& \frac{0.00047(\theta_2 + \theta_3\cos(x_{r3}))x_{r4}\theta_3^2\sin(2x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos x_{r3})^2}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial f_2}{\partial x_4} = {} & \frac{-\theta_2\theta_3\sin(x_{r3})x_{r2}}{-\theta_2\theta_1 + \theta_3^2} - \frac{\theta_2\theta_3\sin(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2} + \frac{\theta_2(-\theta_3\sin(x_{r3})x_{r4} - \theta_3\sin(x_{r3})x_{r2})}{-\theta_2\theta_1 + \theta_3^2} \\
& - \frac{0.00047(\theta_2 + \theta_3\cos(x_{r3}))}{-\theta_2\theta_1 + \theta_3^2}
\end{aligned}
$$

$$
\frac{\partial f_4}{\partial x_1} = \frac{-(\theta_2 + \theta_3\cos(x_{r3}))(-\theta_4 g\sin(x_{r1}) - \theta_5 g\sin(x_{r1} + x_{r3}))}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \frac{(\theta_1 + \theta_2 + 2\theta_3\cos(x_{r3}))\theta_5 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2}
$$

$$
\begin{aligned}
\frac{\partial f_4}{\partial x_2} = {} & \frac{2(\theta_1 + \theta_2 + 2\theta_3\cos(x_{r3}))(\theta_3)\sin(x_{r3})x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \frac{2(\theta_2 + \theta_3\cos(x_{r3}))\theta_3\sin(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \\
& \frac{0.00545(\theta_2 + \theta_3\cos(x_{r3}))\theta_5 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial f_4}{\partial x_3} = {} & \frac{-\theta_3 \sin(x_{r3})T}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \frac{2(\theta_2 + \theta_3\cos(x_{r3}))T\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2} - \\
& \frac{\theta_3^2 \sin(x_{r3})^2 x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \frac{2(\theta_2 + \theta_3\cos(x_{r3}))\theta_3^3(\sin(x_{r3}))^2 x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} \\
& - \frac{(\theta_2 + \theta_3\cos(x_{r3}))\theta_3\cos(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \frac{2\theta_3^2(\sin(x_{r3}))^2 x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} \\
& - \frac{2(\theta_2 + \theta_1 + 2\theta_3\cos(x_{r3}))\theta_3^3(\sin(x_{r3}))^2 x_{r2}\cos(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2} - \\
& \frac{(\theta_1 + \theta_2 + 2\theta_3\cos(x_{r3}))\theta_3\cos(x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \\
& \frac{\theta_3\sin(x_{r3})(-\theta_3\sin(x_{r3})x_{r4} - \theta_3\sin(x_{r3})x_{r2})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \\
& \frac{2(\theta_2 + \theta_3\cos(x_{r3}))(-\theta_3\sin(x_{r3})x_{r4} - \theta_3\sin(x_{r3})x_{r2})x_{r4}\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2} - \\
& \frac{(\theta_2 + \theta_3\cos(x_{r3}))(-\theta_3\cos(x_{r3})x_{r4} - \theta_3\cos(x_{r3})x_{r2})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \\
& \frac{\theta_3\sin(x_{r3})(\theta_4 g\cos(x_{r1}) + \theta_5 g\cos(x_{r1} + x_{r3}))}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \\
& \frac{2(\theta_2 + \theta_3\cos(x_{r3}))(\theta_4 g\cos(x_{r1}) + \theta_5 g\cos(x_{r1} + x_{r3}))\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2} + \\
& \frac{(\theta_2 + \theta_3\cos(x_{r3}))\theta_5 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \frac{2\theta_3\sin(x_{r3})\theta_5 g\cos(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \\
& \frac{2(\theta_1 + \theta_2 + 2\theta_3\cos(x_{r3}))\theta_5 g\cos(x_{r1} + x_{r3})\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2} - \\
& \frac{(\theta_1 + \theta_2 + 2\theta_3\cos(x_{r3}))\theta_5 g\sin(x_{r1} + x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \frac{0.00545\theta_3\sin(x_{r3})x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} - \\
& \frac{0.0109(\theta_2 + \theta_3\cos(x_{r3}))x_{r2}\theta_3^2\cos(x_{r3})\sin(x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2} - \frac{0.00094\theta_3\sin(x_{r3})x_{r4}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \\
& \frac{0.00047(\theta_1 + \theta_2 + 2\theta_3\cos(x_{r3}))x_{r4}\theta_3^2\sin(2x_{r3})}{\left(-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2\right)^2}
\end{aligned}
$$

$$\frac{\partial f_4}{\partial x_4} = \frac{-(\theta_2 + \theta_3 \cos(x_{r3}))\theta_3 \sin(x_{r3})x_{r2}}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2} + \frac{(\theta_2 + \theta_3 \cos(x_{r3}))(-\theta_3 \sin(x_{r3})x_{r4} - \theta_3 \sin(x_{r3})x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2}$$

$$\frac{0.00047(\theta_1 + \theta_2 + 2\theta_3 \cos(x_{r3}))}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2}$$

$$\frac{\partial f_2}{\partial u} = \frac{-\theta_2}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))}$$

$$\frac{\partial f_4}{\partial u} = \frac{\theta_2 + \theta_3 \cos(x_{r3})}{-\theta_2\theta_1 + \theta_3^2(\cos(x_{r3}))^2}$$

APPENDIX B

Maple Files

The Maple Files used to calculate the PI and GAMA for the local linear model (20 0 70 0)

```
> with(linalg):
> A:=matrix(4,4,[
>               0,    1.00000000000000,  0, 0,
> 17.12782670872315,    -0.18286115510401,-6.35240463303645,
0.02060351723154,0,           0,         0, 1.0000000000000
>35.89108022594552,    0.23891312534449 ,   66.56864263132525, -
0.07125868422174]):
> B:=matrix(4,1,[0,
>              33.55250552367133,
>                0,
>              -43.83727070541081]):
> C:=matrix(1,4,[0,0,1.00000000000000,0]):
> PIs:=matrix(4,3,[]):
> GAMA:=matrix(1,3,[]):
> P:=matrix(4,3,0):
> Q:=matrix(1,3,[evalf(70*Pi/180),evalf(-50*Pi/180),0]):
> S:=matrix(3,3,[0,0,0,0,0,.5,0,-.5,0]):
> mat:=evalm(PIs&*S):
> mat1:=evalm(A&*PIs+B&*GAMA+P):
```

```
> mat2:=evalm(C&*PIs):

> mat3:=evalm(-Q):

>eqs:={mat[1,1]=mat1[1,1],mat[1,2]=mat1[1,2],mat[1,3]=mat1[1,3
],mat[2,1]=mat1[2,1],mat[2,2]=mat1[2,2],mat[2,3]=mat1[2,3]
mat[2,3]=mat1[2,3], mat[3,1]=mat1[3,1],mat[3,2]=mat1[3,2],
mat[3,3]=mat1[3,3],
mat[4,1]=mat1[4,1],mat[4,2]=mat1[4,2],mat[4,3]=mat1[4,3],mat2[
1,1]=mat3[1,1],mat2[1,2]=mat3[1,2],mat2[1,3]=mat3[1,3]}:

>eqs2:=solve({eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8,eqn9,eqn
10,eqn11,eqn12,eqn13,eqn14,eqn15},{PIs[1,1],PIs[1,2],PIs[1,3],
PIs[2,1],PIs[2,2],PIs[2,3],PIs[3,1],PIs[3,2],PIs[3,3],PIs[4,1]
,PIs[4,2],PIs[4,3],GAMA[1,1],GAMA[1,2],GAMA[1,3]}):
```

Appendix C

Matlab Files

```
%The Matlab Files used to calculate the gain of the Local Linear Model1(20 0 70 0).

a=[0 1 0 0;17.1278 -0.1829 -6.3524 0.0206;

  0 0 0 1;35.8911 0.2389 66.5686 -0.0713];

b=[0;33.5525;0;-43.8373];

q=[7 0 0 0;0 0 0 0;0 0 7 0;0 0 0 0];

r=[1];

k=lqr(a,b,q,r);

PIs(3,3)= 0.; PIs(2,1) = 0.; PIs(4,1) = 0.; PIs(4,2) = 0.;

PIs(2,3) = -.5229139623; PIs(3,2) = 1.047197551; PIs(3,1) = -1.221730477;

GAMA(1,3) = -.3376614858e-2; GAMA(1,2) = .7399269416; PIs(2,2) = -.1981047810e-3;

GAMA(1,1) = -.8549737765; PIs(1,1) = 1.221730367; PIs(4,3) = .5235987755;

PIs(1,3) = .3962095620e-3; PIs(1,2) = -1.045827925;

PIs=[PIs(1,1) PIs(1,2) PIs(1,3);PIs(2,1) PIs(2,2) PIs(2,3);PIs(3,1) PIs(3,2) PIs(3,3);PIs(4,1)
PIs(4,2) PIs(4,3)];

GAMA=[GAMA(1,1),GAMA(1,2),GAMA(1,3)];

l=GAMA+k*PIs;
```

Appendix D

Local Linear Models and Control Gains

1 Trajectory tracking around the top equilibrium point.

1.1 Local Linear Model

Linear Models for $[20^0\ \ 0\ \ 70^0\ \ 0]$

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 17.1278 & -0.1829 & -6.3524 & 0.0206 \\ 0 & 0 & 0 & 1 \\ 35.8911 & 0.2389 & 66.5686 & -0.0173 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 \\ 33.5525 \\ 0 \\ -43.8373 \end{bmatrix} \quad C_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Models for $[55^0\ \ 0\ \ 35^0\ \ 0]$

$$A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 48.7352 & -0.2172 & -18.0751 & 0.0325 \\ 0 & 0 & 0 & 1 \\ -26.2449 & 0.3767 & 89.6141 & -0.1007 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ 39.8614 \\ 0 \\ 69.1255 \end{bmatrix} \quad C_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Models for $[90^0\ \ 0\ \ 0^0\ \ 0]$

$$A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 66.9725 & -0.2445 & -24.8389 & 0.0400 \\ 0 & 0 & 0 & 1 \\ -68.7259 & 0.4637 & 105.3692 & -0.1202 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 \\ 44.8715 \\ 0 \\ -85.0866 \end{bmatrix} \quad C_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Linear Models for $[125^0\ 0\ -35^0\ 0]$

$$A_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 48.7352 & -0.2172 & -18.0751 & 0.0325 \\ 0 & 0 & 0 & 1 \\ -26.2449 & 0.3767 & 89.6141 & -0.1007 \end{bmatrix} \quad B_4 = \begin{bmatrix} 0 \\ 39.8614 \\ 0 \\ 69.1255 \end{bmatrix} \quad C_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Models for $[160^0\ \ 0\ \ -70^0\ \ 0]$

$$A_5 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 17.1278 & -0.1829 & -6.3524 & 0.0206 \\ 0 & 0 & 0 & 1 \\ 35.8911 & 0.2389 & 66.5686 & -0.0173 \end{bmatrix} \quad B_5 = \begin{bmatrix} 0 \\ 33.5525 \\ 0 \\ -43.8373 \end{bmatrix} \quad C_5 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^{t}$$

1.2. Local Control Gains:

$$\Lambda = \begin{bmatrix} 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad R = 1$$

Linear control gains for 50sin0.5t

Linear model 1: K_1=[68.33 10.11 65.22 8.49]; L_1=[−4.65 3.25 0.68]

Linear model 2: K_2=[31.3 5.45 28.88 3.69]; L_2=[-2.5 3.5 0.75]

Linear model 3: K_3=[26.28 4.87 24.06 3.05]; L_3=[0 3.63 0.77]

Linear model 4: K_4=[31.3 5.45 28.88 3.69]; L_4=[2.5 3.5 0.75]

Linear model 5: K_5=[68.33 10.11 65.22 8.49]; L_5=[4.65 3.25 0.68]

Linear control gains for 60sin0.5t

Linear model 1: K_1=[68.33 10.11 65.22 8.49]; L_1=[−4.65 3.9 0.81]

Linear model 2: K_2=[31.3 5.45 28.88 3.69]; L_2=[-2.5 4.19 0.89]

Linear model 3: K_3=[26.28 4.87 24.06 3.05]; L_3=[0 4.36 0.93]

Linear model 4: K_4=[31.3 5.45 28.88 3.69]; L_4=[2.5 4.19 0.89]

Linear model 5: K_5=[68.33 10.11 65.22 8.49]; L_5=[4.65 3.9 0.81]

Linear control gains for 70sin0.5t

Linear model 1: K_1=[68.33 10.11 65.22 8.49]; L_1=[−4.65 4.55 0.95]

Linear model 2: K_2=[31.3 5.45 28.88 3.69]; L_2=[-2.5 4.89 1.04]

Linear model 3: K_3=[26.28 4.87 24.06 3.05]; L_3=[0 5.09 1.08]

Linear model 4: K_4=[31.3 5.45 28.88 3.69]; L_4=[2.5 4.89 1.04]

Linear model 5: K_5=[68.33 10.11 65.22 8.49]; L_5=[4.65 4.55 0.95]

2. Trajectory Tracking around the mid equilibrium point

2.1 Local Linear Model

Linear Models for [-20^0 0 110^0 0]

$$A_{m1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -17.1278 & -0.1829 & 6.3524 & 0.0109 \\ 0 & 0 & 0 & 1 \\ 70.1467 & 0.1268 & 53.8638 & -0.0519 \end{bmatrix} \quad B_{m1} = \begin{bmatrix} 0 \\ 33.5525 \\ 0 \\ -23.2677 \end{bmatrix} \quad C_{m1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Models for [-55^0 0 145^0 0]

$$A_{m2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -48.7352 & -0.2172 & 18.0751 & 0.005 \\ 0 & 0 & 0 & 1 \\ 71.2255 & 0.0578 & 53.4639 & -0.0457 \end{bmatrix} \quad B_{m2} = \begin{bmatrix} 0 \\ 39.8614 \\ 0 \\ -10.5973 \end{bmatrix} \quad C_{m2} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Models for [-90^0 0 180^0 0]

$$A_{m3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -66.9725 & -0.2445 & 24.8389 & 0.0022 \\ 0 & 0 & 0 & 1 \\ 65.219 & 0.02538 & 55.6914 & -0.0446 \end{bmatrix} \quad B_{m3} = \begin{bmatrix} 0 \\ 44.8715 \\ 0 \\ -4.6565 \end{bmatrix} \quad C_{m3} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Linear Models for [-125^0 0 215^0 0]

$$A_{m4} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -48.7352 & -0.2172 & 18.0751 & 0.005 \\ 0 & 0 & 0 & 1 \\ 71.2255 & 0.0578 & 53.4649 & -0.0457 \end{bmatrix} \quad B_{m4} = \begin{bmatrix} 0 \\ 39.8614 \\ 0 \\ -10.5973 \end{bmatrix} \quad C_{m4} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Models for [-160^0 0 250^0 0]

$$A_{m5} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -17.1278 & -0.1829 & 6.3524 & 0.0109 \\ 0 & 0 & 0 & 1 \\ 70.1467 & 0.1268 & 53.8638 & -0.0519 \end{bmatrix} \quad B_{m5} = \begin{bmatrix} 0 \\ 33.5525 \\ 0 \\ -23.2677 \end{bmatrix} \quad C_{m5} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

2.2. Local Control Gains:

$$\Lambda = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad R = 1$$

Linear control gains for 180+50sin0.5t

Linear model 1: K_{m1}=[35.71 4.2 38.53 5.01]; L_{m1}=[2.59 1.81 0.36]

Linear model 2: K_{m2}=[11 1.1 15.29 1.93]; L_{m2}=[1.59 2.25 0.36]

Linear model 3: K_{m3}=[7.44 0.68 12.35 1.53]; L_{m3}=[0 2.47 0.37]

Linear model 4: K_{m4}=[11 1.1 15.29 1.93]; L_{m4}=[-1.59 2.25 0.36]

Linear model 5: K_{m5}=[35.71 4.2 38.53 5.01]; L_{m5}=[-2.59 1.81 0.36]

Linear control gains for 180+60sin0.5t

Linear model 1: K_{m1}=[35.71 4.2 38.53 5.01]; L_{m1}=[2.59 2.18 0.43]

Linear model 2: K_{m2}=[11 1.1 15.29 1.93]; L_{m2}=[1.59 2.7 0.43]

Linear model 3: K_{m3}=[7.44 0.68 12.35 1.53]; L_{m3}=[0 2.97 0.44]

Linear model 4: K_{m4}=[11 1.1 15.29 1.93]; L_{m4}=[-1.59 2.7 0.36]

Linear model 5: K_{m5}=[35.71 4.2 38.53 5.01]; L_{m5}=[-2.59 1.81 0.43]

Linear control gains for 180+70sin0.5t

Linear model 1: K_{m1}=[35.71 4.2 38.53 5.01]; L_{m1}=[2.59 2.54 0.5]

Linear model 2: K_{m2}=[11 1.1 15.29 1.93]; L_{m2}=[1.59 3.15 0.51]

Linear model 3: K_{m3}=[7.44 0.68 12.35 1.53]; L_{m3}=[0 3.46 0.52]

Linear model 4: K_{m4}=[11 1.1 15.29 1.93]; L_{m4}=[-1.59 3.15 0.51]

Linear model 5: K_{m5}=[35.71 4.2 38.53 5.01]; L_{m5}=[-2.59 2.54 0.5]

Appendix E

Source Code for the Top Trajectory Tracking

```
#define SAMPLE 0.005 /* sample period between .001 and .016 */

#define NUM_POINTS 1000 /* number of data points to write to a data M-file */

#define POINT_INCREMENT 20 /* After POINT_INCREMENT time interval the */

#define PI 3.1415926

#define HALFPI 1.5707963

#define G 9.804 /* acceleration of gravity m/s^2 */

#define INIT_RAD_ENC1 -HALFPI /* Pendubot hanging position initial conditions */

#define INIT_RAD_ENC2 0.0

#include "devproj.h"

/* Beginning of the main program */

void main(argc, argv)

   int argc;

  char *argv[];

{

   /* define and initialize variables here */

   float x1k=0.0;

   float x3k=0.0;

   float x1old = 0.0;

   float x3old = 0.0;

   float x2k=0.0,x4k=0.0;

   float x2old1,x2old2;
```

```
float x4old1,x4old2;

float uf;

float u,t;

char datafile[13];

int         catch;

int           pd;

float Kp=58.0;  /* swing up outer loop control gains */

float Kd=9.55;

float rg=180.0/PI;    /* a variable used to change radiums into degrees*/

float A1a=20.0*HALFPI/90.0,A1b=70.0*HALFPI/90.0;

float A2a=55.0*HALFPI/90.0,A2b=35.0*HALFPI/90.0;

float A3a=90.0*HALFPI/90.0,A3b=0.0*HALFPI/90.0;

float A4a=125.0*HALFPI/90.0,A4b=-35.0*HALFPI/90.0;

float A5a=160.0*HALFPI/90.0,A5b=-70.0*HALFPI/90.0;

float una=0.0,unb=0.0, float un=0.0;

float dosa=0.0,dosb=0.0, float dos=0.0;

float tresa=0.0,tresb=0.0, float tres=0.0;

float cuatroa=0.0,cuatrob=0.0, float cuatro=0.0;

float cincoa=0.0,cincob=0.0, float cinco=0.0;

float v6=0.0,v2=0.0, float sum=0.0;

float v3=0.0,v4=0.0;

float v5=0.0, float tn=0.0;

float P[5] = {0.0308,0.0106,0.0095,0.2087,0.0630}; /* identified parameters */
```

```
float d11,d12or21,d22;    /* Partial feedback linearization gains */

float c11,c12,c21;

float phi[2];

float v1,dbar,c1bar,c2bar,gbar,h;

float dither_w=20.0,dither_ampl=0.25; /* dither input to get ride of some friction
                                                    effects*/

float offset=0.0;   /* offset is a open loop voltage added to the balancing control */

float w1=0,w4=0; /*reference signals*/

float w2=0, w3=0; /*reference signals*/

/* Set DAC output to zero initially */

zeroDAC();

/* see devproj.h for a description of pause_message */

pause_message("Make sure the links are at rest and then press any key\n");

/* see devproj.h for a description of init_boards *

init_boards();

x1old = -HALFPI;   /* initialize position and velocity variables */

x3old = 0.0;

x2old1 = 0.0;

x2old2 = 0.0;

x4old1 = 0.0;

x4old2 = 0.0;

t = 0.0;

u = 0.0;
```

```
pd = 1;

catch = 0;

pause_message("Please hold man switch and press any key\n");

printf("Press Any Key to Stop Control\n");

/* start the timer counting */

/* see timerdb.h */

arm_counter();

/* start of the continuous control loop */

do

{

    /* read the position of the links */

    /* see encodrbd.h */

    read_encoders(&x1k,&x3k);

    /* this is a precaution to check if the first link is moving too far */

    /* away from its normal operating area*/

    safety_check(x1k,(float) PI,(float) -PI);

    /* calculate raw velocity */

    x2k = (x1k-x1old)/SAMPLE; /* calculate velocity */

    x4k = (x3k-x3old)/SAMPLE;

    /* filter velocity with an average */

    x2k = (x2k+x2old1+x2old2)/3.0;  /* average last 3 velocities to get */

    x4k = (x4k+x4old1+x4old2)/3.0;  /* ride of quantization noise */

/* The control algorithm that outputs a control effort "u" in the range of + or - 10V */
```

```
    if (!catch) { /* if links have not come in range to balance */
    if (fabs(x1k-HALFPI) < .80) {
      if (fabs(x3k) < .50) {
      u=26.28034864*(x1k-A3a)+4.866475127*(x2k)+24.06187969*(x3k-
         A3b)+3.049104462*x4k;
           /* calculate balancing control */
     if (fabs(u) < 6) { /* if balancing control check is not too large */
      pd = 0;        /* switch from swing up control to balance */
      catch = 1;
     } /* endif */
    } /* endif */
  } /* endif */
} /* endif */
 if (pd) {   /* if swing up control is still in use */
 d11 = P[0] + P[1] + 2*P[2]*cos(x3k);
 d12or21 = P[1] + P[2]*cos(x3k);
 d22 = P[1];                 /* Calculate Lagrangian dynamics */
 h = -P[2]*sin(x3k);
 c11 = h*x4k;
 c12 = h*x4k + h*x2k;
 c21 = -h*x2k;
 phi[0] = P[3]*G*cos(x1k) + P[4]*G*cos(x1k + x3k);
 phi[1] = P[4]*G*cos(x1k + x3k);
```

```
dbar = d11 - (d12or21*d12or21/d22);   /* Partial Feedback Linearization */

c1bar = c11 - (d12or21*c21/d22);

c2bar = c12;

gbar = phi[0] - (d12or21*phi[1]/d22);

/* outer loop control */

v1 = -Kd*(x2k) + Kp*(HALFPI-x1k);

/* inner loop control */

u = dbar*v1 + c1bar*x2k + c2bar*x4k + gbar;

  } else { /* perform fuzzy control */

           w1=1;

           w2=sin(0.5*tn);

           w3=cos(0.5*tn);

           w4=1.22173*w2;

           tn=tn+0.005;

           if(x1k*rg>-15.0 && x1k*rg<20.0){

              una=(x1k*rg+15.0)/35.0;

           }else{

            una=0.0;

           }

           if(x1k*rg>20.0 && x1k*rg<55.0){

              unb=((20.0-x1k*rg)/35.0)+1.0;

           }else{

            unb=0.0;
```

```
}
if (una>=unb){
   un=una;
}else{
 un=unb;
}
if (x1k*rg>20.0 && x1k*rg<55.0){
   dosa=(x1k*rg-20.0)/35.0;
}else{
 dosa=0.0;
}
if (x1k*rg>55.0 && x1k*rg<90.0){
   dosb=((55.0-x1k*rg)/35.0)+1.0;
}else{
 dosb=0.0;
}
if (dosa>=dosb){
   dos=dosa;
}else{
 dos=dosb;
}
if (x1k*rg>55.0 && x1k*rg<90.0){
   tresa=(x1k*rg-55.0)/35.0;
```

```
}else{
 tresa=0.0;
}
if(x1k*rg>90.0 && x1k*rg<125.0){
   tresb=((90.0-x1k*rg)/35.0)+1.0;
}else{
 tresb=0.0;
}
if(tresa>=tresb){
   tres=tresa;
}else{
 tres=tresb;
}
if(x1k*rg>90.0 && x1k*rg<125.0){
   cuatroa=(x1k*rg-90.0)/35.0;
}else{
 cuatroa=0.0;
}
if(x1k*rg>125.0 && x1k*rg<160.0){
   cuatrob=((125.0-x1k*rg)/35.0)+1.0;
}else{
 cuatrob=0.0;
}
```

```
if (cuatroa>=cuatrob){
   cuatro=cuatroa;
}else{
 cuatro=cuatrob;
}
if (x1k*rg>125.0 && x1k*rg<160.0){
   cincoa=(x1k*rg-125.0)/35.0;
}else{
 cincoa=0.0;
}
if (x1k*rg>160.0 && x1k*rg<195.0){
   cincob=((160.0-x1k*rg)/35.0)+1.0;
}else{
 cincob=0.0;
}
if (cincoa>=cincob){
   cinco=cincoa;
}else{
 cinco=cincob;
}
sum=un+dos+tres+cuatro+cinco;
v6=68.3316*(x1k-A1a)+10.1064*(x2k)+65.22486*(x3k-A1b)+8.48716*x4k
   +1.9227-4.65056*w1+4.552*w2+0.94552*w3;
```

```
v2=31.29983*(x1k-A2a)+5.45113*(x2k)+28.87608*(x3k- A2b)+ 3.6949*x4k
    +1.1736-2.50449*w1+4.89365*w2+1.04359*w3;
v3=26.28035*(x1k-A3a)+4.86648*(x2k)+24.06188*(x3k-A3b)+ 3.0491*x4k
  +5.08863*w2+1.08216*w3;
v4=31.29983*(x1k-A4a)+5.45113*(x2k)+28.87608*(x3k-A4b)+3.6949*x4k
   -1.1736+2.50449*w1+4.89365*w2+1.04359*w3;
v5=68.3316*(x1k-A5a)+10.1064*(x2k)+65.22486*(x3k-A5b)+8.48716*x4k
   -1.9227+4.65056*w1+4.551993*w2+0.945522*w3;
    uf=(v6*un+v2*dos+v3*tres+v4*cuatro+v5*cinco)/sum;
    u=uf;
    if (fabs(x1k-HALFPI) < .2) {
        u = u + dither_ampl*sin(dither_w*t) + offset; /* add dither signal */
     } /* endif */
     if (fabs(u) > 6) { /* if control gets too large switch back to */
                    /* swing up control which is servoing around */
                    /* a set point */
        u = 0.0; /* set control to zero */
        pd = 1;
     } /* endif */
    } /* endif */
  /* limit output */
    if (u > 9.95)
     u = 9.95;
```

```
if (u < -9.95)
  u = -9.95;
 /* output control effort */
/* see dacbd.h  */
out_DAC0(u);
/* keep track of past states */
x2old2 = x2old1;  /* save old positions and velocities */
x2old1 = x2k;
x4old2 = x4old1;
x4old1 = x4k;
x1old = x1k;
x3old = x3k;
 /* save up to 8 variables */
  /* see devproj.h */
   savedata(t,x1k,x3k,w4,u,0.0,0.0,0.0); /* savedata needs 8 pars */
 /* that is the reason for the zeros */
/* increment time */
t = t+SAMPLE;
/* I use this to watch if a correct sample rate is being produced */
/* This routine toggles DIO output pin 0 each sample period.  You */
/* can scope the output pin to monitor the sample rate created */
/* see timerbd.h */
UpdateDIO_Output_Pin0();
```

```
    /* This function wait for the sample period to expire before returning */
    /* See timerbd.h */
    WaitForSample();
  }
  /* Continue controlling until any key is hit */
  while (!kbhit()); /* continue control until any key hit */
  /* send zero to the DAC output */
  /* see dacbd.h */
  zeroDAC();
  /* stop counter */
  /* see timerbd.h */
  disarm_counter();
  /* write saved data to the filename specified at the command line *
  /* see devproj.h */
  savedatafile(datafile);
  /* write saved data to the file "dontremv.m" used in the Matlab plotting command */
  /* see devproj.h */
  savedataforMatlabplotting("c:\\matlab\\");
  /* Free the memory allocated to save data */
  /* see devproj.h */
  myfree();
} /* end of program */
```

Appendix F

Source Code for the Mid Trajectory Tracking

```
#define SAMPLE 0.005 /* sample period between .001 and .016 */

#define NUM_POINTS 1000 /* number of data points to write to a data M-file */

#define POINT_INCREMENT 20 /* After POINT_INCREMENT time interval the */

#define PI 3.1415926

#define HALFPI 1.5707963

#define G 9.804 /* acceleration of gravity m/s^2 */

#define INIT_RAD_ENC1 -HALFPI /* Pendubot hanging position initial conditions */

#define INIT_RAD_ENC2 0.0

#include "devproj.h"

/* Beginning of the main program */

void main(argc, argv)

   int argc;

  char *argv[];

{

   /* define and initialize variables here */

   float x1k=0.0;

   float x3k=0.0;

   float x1old = 0.0;

   float x3old = 0.0;

   float x2k=0.0,x4k=0.0;

   float x2old1,x2old2;
```

```
float x4old1,x4old2;
float uf;
float u,t;
char datafile[13];
int        catch;
int          pd;
float Kp=240.0; /* swing up outer loop control gains */
float Kd=10.0;
float P[5] = {0.0308,0.0106,0.0095,0.2087,0.0630}; /* identified parameters */
float d11,d12or21,d22;   /* Partial feedback linearization gains */
float c11,c12,c21;
float phi[2];
float v1,dbar,c1bar,c2bar,gbar,h;
float q1d[3],ts;
float w=4.5,ampl=1.165;  /* swing up trajectory u=ampl*sin(w*t) */
float rg=180.0/PI; /*a variable used to change radiums into degrees*/
float A1a=-20.0*HALFPI/90.0, A1b=110.0*HALFPI/90.0;
float A2a=-55.0*HALFPI/90.0, A2b=145.0*HALFPI/90.0;
float A3a=-90.0*HALFPI/90.0, A3b=180.0*HALFPI/90.0;
float A4a=-125.0*HALFPI/90.0, A4b=215.0*HALFPI/90.0;
float A5a=-160.0*HALFPI/90.0, A5b=250.0*HALFPI/90.0;
float xs=0.0;
float una=0.0,unb=0.0, un=0.0;
```

```
float dosa=0.0,dosb=0.0, dos=0.0;
float tresa=0.0,tresb=0.0, tres=0.0;
float cuatroa=0.0,cuatrob=0.0, cuatro=0.0;
float cincoa=0.0,cincob=0.0, cinco=0.0;
float sum=0.0, l=0.0;
float v6=0.0,v2=0.0;
float v3=0.0,v4=0.0;
float v5=0.0;
float w1=0, w2=0,uf=0.0; /*reference signals*/
float w3=0,w4=0,tn=0.0; /*reference signals*/
/* Set DAC output to zero initially */
zeroDAC();
/* see devproj.h for a description of pause_message */
pause_message("Make sure the links are at rest and then press any key\n");
/* see devproj.h for a description of init_boards */
init_boards();
x1old = -HALFPI;  /* initialize position and velocity variables */
x3old = 0.0;
x2old1 = 0.0;
x2old2 = 0.0;
x4old1 = 0.0;
x4old2 = 0.0;
t = 0.0;
```

```
u = 0.0;
pd = 1;
catch = 0;
ts = 2*PI/w; /* time in swing up control that the trajectory is changed */
        /* from a sinusoidal to a step */
pause_message("Please hold man switch and press any key\n");
printf("Press Any Key to Stop Control\n");
/* start the timer counting */
/* see timerdb.h */
arm_counter( );
/* start of the continuous control loop */
do
{
  /* read the position of the links */
  /* see encodrbd.h */
  read_encoders(&x1k,&x3k);
  /* this is a precaution to check if the first link is moving too far */
  /* away from its normal operating area*/
  safety_check(x1k,(float) PI,(float) -PI);
  /* calculate raw velocity */
  x2k = (x1k-x1old)/SAMPLE; /* calculate velocity */
  x4k = (x3k-x3old)/SAMPLE;
  /* filter velocity with an average */
```

```
x2k = (x2k+x2old1+x2old2)/3.0;  /* average last 3 velocities to get */
x4k = (x4k+x4old1+x4old2)/3.0;  /* ride of quantization noise */
/* The control algorithm that outputs a control effort "u" in the range of + or - 10V */
    if (!catch) { /* if links have not come in range to balance */
    if (fabs(x1k-HALFPI) < .80) {
     if (fabs(x3k) < .50) {
     u=26.28034864*(x1k-A3a)+4.866475127*(x2k)+24.06187969*(x3k-
        A3b)+3.049104462*x4k;
          /* calculate balancing control */
     if (fabs(u) < 6) { /* if balancing control check is not too large */
      pd = 0;          /* switch from swing up control to balance */
      catch = 1;
     } /* endif */
    } /* endif */
   } /* endif */
  } /* endif */
  if (pd) {   /* if swing up control is still in use */
  d11 = P[0] + P[1] + 2*P[2]*cos(x3k);
  d12or21 = P[1] + P[2]*cos(x3k);
  d22 = P[1];                /* Calculate Lagrangian Dynamics*/
  h = -P[2]*sin(x3k);
  c11 = h*x4k;
  c12 = h*x4k + h*x2k;
```

```
c21 = -h*x2k;
phi[0] = P[3]*G*cos(x1k) + P[4]*G*cos(x1k + x3k);
phi[1] = P[4]*G*cos(x1k + x3k);
dbar = d11 - (d12or21*d12or21/d22);   /* Partial Feedback Linearization */
c1bar = c11 - (d12or21*c21/d22);
c2bar = c12;
gbar = phi[0] - (d12or21*phi[1]/d22);
 if (t<ts) {  /* Trajectory for Pendubot to follow for swing up */
  q1d[0] = ampl*sin(w*t)-HALFPI;
  q1d[1] = w*ampl*cos(w*t);
  q1d[2] = -w*w*ampl*sin(w*t);
} else {
  q1d[0] = -HALFPI;
  q1d[1] = 0.0;
  q1d[2] = 0.0;
} /* endif */
/* outer loop control *
v1 = q1d[2] + Kd*(q1d[1] - x2k) + Kp*(q1d[0]-x1k);
/* inner loop control *
u = dbar*v1 + c1bar*x2k + c2bar*x4k + gbar;
  } else { /* perform fuzzy control */
  w1=1;
  w2=sin(0.5*tn);
```

```
w3=cos(0.5*tn);
w4=PI+1.2217305*w2;
tn=tn+0.005;
xs=-1*x1k;
if(xs*rg>-15.0 && xs*rg<20.0){
        una=(xs*rg+15.0)/35.0;
}else{
        una=0.0;
}
if(xs*rg>20.0 && xs*rg<55.0){
        unb=((20.0-xs*rg)/35.0)+1.0;
}else{
        unb=0.0;
}
if(una>=unb){
            un=una;
}else{
        un=unb;
}
if(xs*rg>20.0 && xs*rg<55.0){
        dosa=(xs*rg-20.0)/35.0;
}else{
        dosa=0.0;
```

```
}
if(xs*rg>55.0 && xs*rg<90.0){
        dosb=((55.0-xs*rg)/35.0)+1.0;
}else{
        dosb=0.0;
}
if(dosa>=dosb){
          dos=dosa;
}else{
 dos=dosb;
        }
if(xs*rg>55.0 && xs*rg<90.0){
        tresa=(xs*rg-55.0)/35.0;
}else{
        tresa=0.0;
}
if(xs*rg>90.0 && xs*rg<125.0){
        tresb=((90.0-xs*rg)/35.0)+1.0;
}else{
        tresb=0.0;
}
if(tresa>=tresb){
        tres=tresa;
```

```
}else{
        tres=tresb;
}
if (xs*rg>90.0 && xs*rg<125.0){
        cuatroa=(xs*rg-90.0)/35.0;
}else{
        cuatroa=0.0;
}
if (xs*rg>125.0 && xs*rg<160.0){
        cuatrob=((125.0-xs*rg)/35.0)+1.0;
}else{
        cuatrob=0.0;
}
if (cuatroa>=cuatrob){
        cuatro=cuatroa;
}else{
        cuatro=cuatrob;
}
if (xs*rg>125.0 && xs*rg<160.0){
        cincoa=(xs*rg-125.0)/35.0;
}else{
        cincoa=0.0;
}
```

```
if (xs*rg>160.0 && xs*rg<195.0){
        cincob=((160.0-xs*rg)/35.0)+1.0;
}else{
        cincob=0.0;
}
if (cincoa>=cincob){
        cinco=cincoa;
}else{
        cinco=cincob;
}
sum=un+dos+tres+cuatro+cinco;
v6=-35.7129*(x1k-A1a)-4.20381*(x2k)-38.5316*(x3k-A1b)- 5.0054*x4k
   +1.9227+2.58872*w1+2.53883*w2+0.49954*w3;
v2=-11.0008*(x1k-A2a)-1.09806*(x2k)-15.28635*(x3k-A2b)-1.9281*x4k
   +1.17359+1.59402*w1+3.14849*w2+0.50719*w3;
v3=-7.44399*(x1k-A3a)-0.67757*(x2k)-12.3513*(x3k-A3b)- 1.52851*x4k
   +3.46033*w2+0.51899*w3;
v4=-11.0008*(x1k-A4a)-1.09806*(x2k)-15.28635*(x3k-A4b)-1.9281*x4k-
   +1.17359-1.59402*w1+3.14849*w2+0.50719*w3;
v5=-35.7129*(x1k-A5a)-4.20381*(x2k)-38.5316*(x3k-A5b)+ 5.0054*x4k
   - 1.9227-2.58872*w1+2.53883*w2+0.49954*w3;
 uf=(v6*un+v2*dos+v3*tres+v4*cuatro+v5*cinco)/sum;
  u=uf;
```

```
    uf=(v6*un+v2*dos+v3*tres+v4*cuatro+v5*cinco)/sum;
    u=uf;
  if (fabs(x1k-HALFPI) < .2) {
      u = u + dither_ampl*sin(dither_w*t) + offset; /* add dither signal */
   } /* endif */
   if (fabs(u) > 6) { /* if control gets too large switch back to */
                /* swing up control which is servoing around */
                /* a set point */
     u = 0.0; /* set control to zero */
     pd = 1;
   } /* endif */
  } /* endif */
/* limit output */
  if (u > 9.95)
   u = 9.95;
  if (u < -9.95)
    u = -9.95;
   /* output control effort */
  /* see dacbd.h  */
  out_DAC0(u);
  /* keep track of past states */
  x2old2 = x2old1;  /* save old positions and velocities */
  x2old1 = x2k;
```

```
  x4old2 = x4old1;
  x4old1 = x4k;
  x1old = x1k;
  x3old = x3k;
  /* save up to 8 variables */
   /* see devproj.h */
    savedata(t,x1k,x3k,w4,u,0,0,0,0,0,0); /* savedata needs 8 pars */
  /* that is the reason for the zeros */
  /* increment time */
  t = t+SAMPLE;
  /* I use this to watch if a correct sample rate is being produced */
  /* This routine toggles DIO output pin 0 each sample period.  You */
  /* can scope the output pin to monitor the sample rate created */
  /* see timerbd.h */
  UpdateDIO_Output_Pin0();
  /* This function wait for the sample period to expire before returning */
  /* See timerbd.h */
  WaitForSample();
}
/* Continue controlling until any key is hit */
while (!kbhit()); /* continue control until any key hit */
/* send zero to the DAC output */
/* see dacbd.h */
```

```
  zeroDAC();
  /* stop counter */
  /* see timerbd.h */
  disarm_counter();
  /* write saved data to the filename specified at the command line */
  /* see devproj.h */
  savedatafile(datafile);
  /* write saved data to the file "dontremv.m" used in the Matlab plotting command */
  /* see devproj.h */
  savedataforMatlabplotting("c:\\matlab\\");
  /* Free the memory allocated to save data */
  /* see devproj.h */
  myfree();
} /* end of program */
```

Appendix G

Local Linear Models and Observers for Simulation

1. Local Linear Models

Linear Model at [-40^0 0 130^0 0]

$$A_{e1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -35.1653 & -0.1998 & 13.0422 & 0.0073 \\ 0 & 0 & 0 & 1 \\ 73.1672 & 0.0847 & 52.7403 & -0.0474 \end{bmatrix} \quad B_{e1} = \begin{bmatrix} 0 \\ 36.654 \\ 0 \\ -15.5382 \end{bmatrix} \quad C_{e1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Model at [-65^0 0 155^0 0]

$$A_{e2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -56.8205 & -0.2289 & 21.0737 & 0.0037 \\ 0 & 0 & 0 & 1 \\ 68.9367 & 0.043 & 54.3126 & -0.045 \end{bmatrix} \quad B_{e2} = \begin{bmatrix} 0 \\ 42.0053 \\ 0 \\ -7.8862 \end{bmatrix} \quad C_{e2} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Model at [-90^0 0 180^0 0]

$$A_{e3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -66.9725 & -0.2445 & 24.8389 & 0.0022 \\ 0 & 0 & 0 & 1 \\ 65.219 & 0.0254 & 55.6914 & -0.0447 \end{bmatrix} \quad B_{e3} = \begin{bmatrix} 0 \\ 44.8715 \\ 0 \\ -4.6565 \end{bmatrix} \quad C_{e3} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Model at [-115^0 0 205^0 0]

$$A_{e4} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -56.8205 & -0.2289 & 21.0737 & 0.0037 \\ 0 & 0 & 0 & 1 \\ 68.9367 & 0.043 & 54.3126 & -0.045 \end{bmatrix} \quad B_{e4} = \begin{bmatrix} 0 \\ 42.0053 \\ 0 \\ -7.8862 \end{bmatrix} \quad C_{e4} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Linear Model at [-140^0 0 230^0 0]

$$A_{cs} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -35.1653 & -0.1998 & 13.0422 & 0.0073 \\ 0 & 0 & 0 & 1 \\ 73.1672 & 0.0847 & 52.7403 & -0.0474 \end{bmatrix} \quad B_{cs} = \begin{bmatrix} 0 \\ 36.654 \\ 0 \\ -15.5382 \end{bmatrix} \quad C_{cs} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

2. Linear Observers

Linear Model 1:

$$G_1 = \begin{bmatrix} 656.614 \\ 6896.709 \\ 769.338 \\ 5251.73 \\ 590.118 \\ 162.044 \\ 1248.168 \end{bmatrix}$$

$$F_1 = \begin{bmatrix} 0 & 1 & -656.614 & 0 & 573.004 & 573.004 & 0 \\ -486.264 & -52.071 & -7404.601 & -66.323 & 6037.388 & 6036.551 & 6.324 \\ 0 & 0 & -769.338 & 1 & 671.374 & 671.374 & 0 \\ 264.405 & 22.074 & -4978.157 & 28.071 & 4574.998 & 4575.353 & -2.681 \\ 0 & 0 & -590.118 & 0 & 514.975 & 514.975 & 0 \\ 0 & 0 & -162.044 & 0 & 141.41 & 141.41 & 0.5 \\ 0 & 0 & -1248.168 & 0 & 1089.232 & 1088.732 & 0 \end{bmatrix}$$

$$H_1 = [-12.3069 \quad -1.4152 \quad -14.2122 \quad -1.8096 \quad 0.5149 \quad 0.4921 \quad 0.1725]$$

Linear Model 2:

$$G_2 = \begin{bmatrix} 915.39 \\ 6886.041 \\ 582.793 \\ 5404.504 \\ 870.782 \\ 98.338 \\ 914.974 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 0 & 1 & -915.39 & 0 & 399.415 & 798.829 & 0 \\ -339.64 & -32.813 & 7247.309 & -47.176 & 3017.744 & 6021.889 & 3.799 \\ 0 & 0 & -582.793 & 1 & 254.917 & 508.593 & 0 \\ 122.034 & 6.16 & -5278.41 & 8.813 & 2355.697 & 4713.941 & -0.713 \\ 0 & 0 & -870.782 & 0 & 379.951 & 759.901 & 0 \\ 0 & 0 & -98.338 & 0 & 42.908 & 85.816 & 0.5 \\ 0 & 0 & -914.974 & 0 & 399.234 & 797.966 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} -6.7329 & -0.7757 & -9.1022 & -1.1232 & 0.3127 & 0.3019 & 0.0904 \end{bmatrix}$$

Linear Model 3:

$$G_3 = \begin{bmatrix} 1021.351 \\ 6055.511 \\ 447.954 \\ 5153.56 \\ 379.692 \\ 475.776 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} 0 & 1 & -1021.351 & 0 & 891.297 & 0 \\ -306.01 & -28.396 & 6383.149 & -42.785 & 5302.492 & 6.332 \\ 0 & 0 & -447.954 & 1 & 390.914 & 0 \\ 90.025 & 2.947 & -5061.291 & 4.396 & 4495.456 & -0.6571 \\ 0 & 0 & -379.692 & 0 & 331.344 & 0.5 \\ 0 & 0 & -475.776 & 0 & 414.693 & 0 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} -5.3272 & -0.6274 & -7.8552 & -0.9535 & 0.4025 & 0.1411 \end{bmatrix}$$

Linear Model 4:

$$G_4 = \begin{bmatrix} 915.39 \\ 6886.041 \\ 582.793 \\ 5404.504 \\ -870.782 \\ 98.338 \\ 914.974 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 0 & 1 & -915.39 & 0 & -399.415 & 798.829 & 0 \\ -339.64 & -32.813 & 7247.309 & -47.176 & -3015.419 & 6028.43 & 6.33 \\ 0 & 0 & -582.793 & 1 & -254.292 & 508.583 & 0 \\ 122.034 & 6.16 & -5278.41 & 8.813 & -2356.134 & 4712.71 & -1.188 \\ 0 & 0 & 870.782 & 0 & 379.951 & -759.901 & 0 \\ 0 & 0 & -98.338 & 0 & -42.908 & 85.816 & 0.5 \\ 0 & 0 & -914.975 & 0 & -399.234 & 797.966 & 0 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} -6.7329 & -0.7757 & -9.1022 & -1.1232 & -0.3127 & 0.3019 & 0.0904 \end{bmatrix}$$

Linear Model 5:

$$G_5 = \begin{bmatrix} 656.614 \\ 6896.709 \\ 769.338 \\ 5251.73 \\ -590.118 \\ 162.044 \\ 1248.168 \end{bmatrix}$$

$$F_5 = \begin{bmatrix} 0 & 1 & -656.614 & 0 & -573.004 & 573.004 & 0 \\ -486.264 & -52.071 & -7404.601 & -66.323 & -6042.785 & 6040.251 & 6.327 \\ 0 & 0 & -769.338 & 1 & -671.374 & 671.374 & 0 \\ 264.405 & 22.074 & -4978.157 & 28.071 & -4572.71 & 4573.785 & -2.682 \\ 0 & 0 & 590.118 & 0 & 514.975 & -514.975 & 0 \\ 0 & 0 & -162.044 & 0 & -141.41 & 141.41 & 0.5 \\ 0 & 0 & -1248.168 & 0 & -1089.232 & 1088.732 & 0 \end{bmatrix}$$

$$H_5 = \begin{bmatrix} -12.3069 & -1.4152 & -14.2122 & -1.8096 & -0.5149 & 0.4921 & 0.1725 \end{bmatrix}$$

Appendix H

Matlab Files

```
%The Matlab Files used to calculate the observers of linear % model (-40 0 130 0)

a=[0 1 0 0 0 0 0;-35.1653 -0.1998 13.0422 0.0073 0 0 0;
   0 0 0 1 0 0 0;73.1762  0.0847 52.7403 -0.0474 0 0 0;
   0 0 0 0 0 0 0;0 0 0 0 0 0 0.5;0 0 0 0 0 -0.5 0];

c=[0 0 1 0 -0.87266463 -0.87266463 0];

p=[-17.5 -35.5 -18 -38.5 -0.3 -2 -1.4];

g=place(a',c',p)'

an=[0 1 0 0;-35.1653 -0.1998 13.0422 0.0073;
    0 0 0 1;73.1762 0.0847 52.7403 -0.0474];

bn=[0;36.654;0;-15.5382];

p1=[-5.2 -6.8 -5.4 -6.6];

cn=[0 0 1 0];

pn=[0 0 0;0 0 0;0 0 0;0 0 0];

q=[-0.87266463 -0.872664626 0];

k=place(an,bn,p1);

s=[0 0 0;0 0 0.5;0 -0.5 0];

PIs(4,3) = .4363323131; PIs(4,2) = 0.; PIs(3,3) = 0.; PIs(4,1) = 0.; PIs(2,1) = 0.; PIs(1,2) = -
.8748176331; PIs(1,3) = .3314226682e-3; PIs(2,3) = -.4374088166; GAMA(1,3) = -
.2155111688e-2; GAMA(1,2) = -1.143832082;

PIs(2,2) = -.1657113341e-3; PIs(1,1) = -.8726646260; GAMA(1,1) = -1.147732343;

PIs(3,2) = .8726646262; PIs(3,1) = .8726646262;
```

```
PIs=[PIs(1,1) PIs(1,2) PIs(1,3);PIs(2,1) PIs(2,2) PIs(2,3);PIs(3,1) PIs(3,2) PIs(3,3);PIs(4,1)
PIs(4,2) PIs(4,3)];
GAMA=[GAMA(1,1) GAMA(1,2) GAMA(1,3)];
l=GAMA+k*PIs;
g0=[g(1,1);g(2,1);g(3,1);g(4,1)];
g1=[g(5,1);g(6,1);g(7,1)];
f11=an-g0*cn+bn*(-k);
f12=pn-g0*q+bn*l;
f21=-g1*cn;
f22=s-g1*q;
f=[f11 f12;f21 f22]
h=[k(1,1) k(1,2) k(1,3) k(1,4) l(1,1) l(1,2) l(1,3)]
```

Appendix I

Source Code for Simulation

```
#define SAMPLE 0.001 /* sample period between .001 and .016 */

#define NUM_COMMANDLINE_PAR 5 /* here enter the number of variables you want
                                 to get at the command line */

#define NUM_POINTS 1000  /* number of data points to write to a data M-file */

#define POINT_INCREMENT 20  /* After POINT_INCREMENT time interval the */
                            /* the savedata function will save a data point */

#define PI 3.1415926

#define HALFPI 1.5707963

#define G 9.804     * acceleration of gravity m/s^2 */

{

   float P[5] = {0.0308,0.0106,0.0095,0.2087,0.0630}; /* identified parameters */

   float g1=0,g2=0,g3=0,g4=0,g5=0,g6=1;

   float dg1=0,dg2=0,dg3=0,dg4=0,dg5=0,dg6=0;

   float xp1=-HALFPI,xp2=0,xp3=PI,xp4=0,dxp1=0,dxp2=0,dxp3=0,dxp4=0,tau=0.0;

   float d11p=0.0,d12p=0.0,d21p=0.0,d22p=0.0;

   float c112p=0.0,c121p=0.0,c211p=0.0,c221p=0.0;

   float phi1p=0.0, phip=0.0;

   float h1p=0.0,detp=0.0;

   float h2p=0.0,y=0, hp=0.0;

   float rg=180.0/PI;

   float A1a=-40.0*HALFPI/90.0,A1b=130.0*HALFPI/90.0;
```

```
float A2a=-65.0*HALFPI/90.0,A2b=155.0*HALFPI/90.0;

float A3a=-90.0*HALFPI/90.0,A3b=180.0*HALFPI/90.0;

float A4a=-115.0*HALFPI/90.0,A4b=205.0*HALFPI/90.0;

float A5a=-140.0*HALFPI/90.0,A5b=230.0*HALFPI/90.0;

float xs=0.0;

float una=0.0,unb=0.0, un=0.0;

float dosa=0.0,dosb=0.0, dos=0.0;

float tresa=0.0,tresb=0.0, tres=0.0;

float cuatroa=0.0,cuatrob=0.0, cuatro=0.0;

float cincoa=0.0,cincob=0.0, cinco=0.0;

float sum=0.0, l=0.0;

float v6=0.0,v2=0.0;

float v3=0.0,v4=0.0;

float v5=0.0;

float w1=1, w2=0,r=0.0,tf=0.0;

float w3=1,w4=0,tn=0.0;

float e1=xp1-A1a,e2=0,e3=xp3-A1b, e4=0, e5=1, e6=0, e7=1;

float de1=0,de2=0,de3=0,de4=0,de5=0,de6=0,de7=0;

float f1=xp1-A2a,f2=0,f3=xp3-A2b, f4=0, f5=1, f6=0,f7=1;

float df1=0,df2=0,df3=0,df4=0,df5=0,df6=0,df7=0;

float h1=xp1-A4a,h2=0,h3=xp3-A4b,h4=0,h5=1,h6=0,h7=1;

float dh1=0,dh2=0,dh3=0,dh4=0,dh5=0,dh6=0,dh7=0;

float i1=xp1-A5a,i2=0,i3=xp3-A5b, i4=0.0,i5=1,i6=0,i7=1;
```

```
float di1=0,di2=0,di3=0,di4=0,di5=0,di6=0,di7=0

/* start the timer counting */

/* see timerdb.h */

arm_counter();

/* start of the continuous control loop */

do

{

      xp1=dxp1*0.001+xp1;

      xp2=dxp2*0.001+xp2;

      xp3=dxp3*0.001+xp3;

      xp4=dxp4*0.001+xp4;

      e1=de1*0.001+e1;

      e2=de2*0.001+e2;

      e3=de3*0.001+e3;

      e4=de4*0.001+e4;

      e5=de5*0.001+e5;

      e6=de6*0.001+e6;

      e7=de7*0.001+e7;

      f1=df1*0.001+f1;

      f2=df2*0.001+f2;

      f3=df3*0.001+f3;

      f4=df4*0.001+f4;

      f5=df5*0.001+f5;
```

```
f6=df6*0.001+f6;
f7=df7*0.001+f7;
g1=dg1*0.001+g1;
g2=dg2*0.001+g2;
g3=dg3*0.001+g3;
g4=dg4*0.001+g4;
g5=dg5*0.001+g5;
g6=dg6*0.001+g6;
h1=dh1*0.001+h1;
h2=dh2*0.001+h2;
h3=dh3*0.001+h3;
h4=dh4*0.001+h4;
h5=dh5*0.001+h5;
h6=dh6*0.001+h6;
h7=dh7*0.001+h7;
i1=di1*0.001+i1;
i2=di2*0.001+i2;
i3=di3*0.001+i3;
i4=di4*0.001+i4;
i5=di5*0.001+i5;
i6=di6*0.001+i6;
i7=di7*0.001+i7;
w1=1;
```

```
w2=sin(0.5*tn);

w3=cos(0.5*tn);

r=xp3-0.872664626*w2-PI;

w4=0.872664626*w2+PI;

tn=tn+0.001;

d11p=P[0]+P[1]+2*P[2]*cos(xp3);

d12p=P[1]+P[2]*cos(xp3);

d21p=d12p;

d22p=P[1];

hp=-1*P[2]*sin(xp3);

c121p=hp;

c211p=hp;

c221p=hp;

c112p=-1*hp;

phi1p=P[3]*G*cos(xp1)+P[4]*G*cos(xp1+xp3);

phi2p=P[4]*G*cos(xp1+xp3);

xs=-1*xp1;

if (xs*rg>15.0 && xs*rg<40.0){

        una=(xs*rg-9.0)/25.0;

}else{

        una=0.0;

}

if (xs*rg>40.0 && xs*rg<65.0){
```

```
        unb=((40.0-xs*rg)/25.0)+1.0;
}else{
        unb=0.0;
}
if (una>=unb){
        un=una;
}else{
        un=unb;
}
if (xs*rg>40.0 && xs*rg<65.0){
        dosa=(xs*rg-40.0)/25.0;
}else{
        dosa=0.0;
}
if (xs*rg>65.0 && xs*rg<90.0){
        dosb=((65.0-xs*rg)/25.0)+1.0;
}else{
        dosb=0.0;
}
if (dosa>=dosb){
        dos=dosa;
}else{
        dos=dosb;
```

```
	}
if (xs*rg>65.0 && xs*rg<90.0){
	tresa=(xs*rg-65.0)/27.0;
}else{
	tresa=0.0;
}
if (xs*rg>90.0 && xs*rg<115.0){
	tresb=((90.0-xs*rg)/25.0)+1.0;
}else{
	tresb=0.0;
}
if (tresa>=tresb){
	tres=tresa;
}else{
	tres=tresb;
}
if (xs*rg>90.0 && xs*rg<115.0){
	cuatroa=(xs*rg-90.0)/25.0;
}else{
	cuatroa=0.0;
}
if (xs*rg>115.0 && xs*rg<140.0){
	cuatrob=((115.0-xs*rg)/25.0)+1.0;
```

```
}else{
        cuatrob=0.0;
}
if (cuatroa>=cuatrob){
        cuatro=cuatroa;
}else{
        cuatro=cuatrob;
}
if (xs*rg>115.0 && xs*rg<140.0){
        cincoa=(xs*rg-115.0)/25.0;
}else{
        cincoa=0.0;
}
if (xs*rg>140.0 && xs*rg<165.0){
        cincob=((140.0-xs*rg)/25.0)+1.0;
}else{
        cincob=0.0;
}
if (cincoa>=cincob){
        cinco=cincoa;
}else{
        cinco=cincob;
}
```

```
sum=un+dos+tres+cuatro+cinco;

v6=-12.30695*(e1)-1.41516*(e2)-14.21221*(e3)-1.80963*e4+0.51492*e5
      +0.49209*e6+0.17252*e7;

v2=-6.73294*(f1)-0.7757*(f2)-9.10221*(f3)-1.12319*f4+0.31275*f5+0.3019*f6
       +0.090432*f7;

v3=-5.32716*(g1)-0.63737*(g2)-7.85525*(g3)+0.95354*g4+0.40252*g5
    +0.14112*g6;

v4=-6.73294*(h1)-0.7757*(h2)-9.10221*(h3)-1.1232*h4-0.31275*f5+0.3019*f6
     +0.090432*f7;

v5=-12.30695*(i1)-1.41516*(i2)-14.21221*(i3)-1.80963*i4+0.51492*i5
       +0.49209*i6+0.17252*i7;

tf=(v6*un+v2*dos+v3*tres+v4*cuatro+v5*cinco)/sum;

tau=tf;

detp=d11p*d22p-d12p*d21p;

 h1p=tau-c211p*xp2*xp4-c121p*xp2*xp4-c221p*xp4*xp4-phi1p-0.00545*xp2;

 h2p=-1*c112p*xp2*xp2-phi2p-0.00047*xp4;

de1=e2-656.614*e3+573.004*e5+573.004*e6+656.614*r;

de2=-486.264*e1-52.071*e2-7404.601*e3+66.323*e4+6037.388*e5
       +6036.551*e6 +6.324*e7+6896.709*r;

de3=-769.338*e3+e4+671.374*e5+671.374*e6+769.338*r;

de4=264.405*e1+22.074*e24978.157*e3+28.071*e4+4574.998*e5+4575.353*e6
       -2.681*e7+5251.73*r;

de5=-590.118*e3+514.975*e5+514.975*e6+590.118*r;
```

de6=-162.044*e3+141.41*e5+141.41*e6+0.5*e7+162.044*r:

de7=-1248.168*e3+1089.232*e5+1088.732*e6+1248.168*r :

df1=1*f2-915.39*f3+399.415*f5+798.829*f6+915.39*r :

df2=-339.64*f1-32.813*f2-7247.309*f3-47.176*f4+3017.744*f5+6021.889*f6

+3.799*f7+6886.041*r :

df3=-582.793*f3+f4+254.292*f5+508.583*f6+582.793*r :

df4=122.034*f1+6.16*f2-5278.41*f3+8.813*f4+2355.697*f5+4713.941*f6

-0.713*f7+5404.504*r:

df5=-870.782*f3+379.951*f5+759.901*f6+870.782*r:

df6=-98.338*f3+42.908*f5+85.816*f6+0.5*f7+98.338*r:

df7=-914.974*f3+399.234*f5+797.966*f6+914.974*r:

dg1=1*g2-1021.351*g3+891.297*g5+1021.351*r:

dg2=-306.01*g1-28.396*g2-6383.149*g3-42.785*g4+5302.492*g5+6.332*g6

-6055.511*r:

dg3=-447.954*g3+g4+390.914*g5+447.954*r:

dg4=-90.025*g1+2.947*g2-5061.291*g3+4.396*g4+4495.456*g5-0.657*g6

+5153.56*r:

dg5=-379.692*g3+331.344*g5+0.5*g6+379.692*r:

dg6=-475.776*g3+414.693*g5+475.776*r:

dh1=1*h2-915.39*h3-399.415*h5+798.829*h6+915.39*r :

dh2=-339.634*h1-32.813*h2-7247.309*h3-47.176*h4-3015.419*h5+6028.43*h6

+6.33*h7+6886.041*r :

dh3=-582.793*h3+f4-254.292*h5+508.583*h6+582.793*r :

```
        dh4=122.034*h1+6.16*h2-5278.41*h3+8.813*h4-2356.137*h5+4712.71*h6
            -0.1.188*h7+5404.504*r;
        dh5=870.782*h3+379.951*h5-759.901*h6-870.782*r;
        dh6=-98.338*h3-42.908*h5+85.816*h6+0.5*f7+98.338*r;
        dh7=-914.975*h3-399.234*h5+797.966*h6+914.975*r;
        di1=i2-656.614*i3-573.004*i5+573.004*i6+656.614*r;
        di2=-486.264*i1-52.071*i2-7404.601*i3+66.323*i4-6042.785*i5 +6040.251*i6
              +6.327*i7+6896.709*r;
        di3=-769.338*i3+i4+671.374*i5+671.374*i6+769.338*r;
        di4=264.405*i1+22.074*i2-4978.157*i3+28.071*i4-4572.71*i5+4573.785*i6
              -2.682*i7+5251.73*r;
        di5=590.118*i3+514.975*i5-514.975*i6-590.118*r;
        di6=-162.044*i3-141.41*i5+141.41*i6+0.5*i7+162.044*r;
        di7=-1248.168*i3-1089.232*i5+1088.732*i6+1248.168*r ;
        dxp1=xp2;
        dxp2=1/detp*(d22p*h1p-d12p*h2p);
        dxp3=xp4;
        dxp4=1/detp*(-1*d21p*h1p+d11p*h2p);
      savedata(t,xp1,xp3,w4,tau,0,0,0,0,0,0); /* savedata needs 8 pars */
                                   /* that is the reason for the zeros */
      t = t+SAMPLE;
  }
} /* end of program */
```